

TK 2000/II ENTENDENDO A ROM GERALDO COEN

TK 2000/II

ENTENDENDO A ROM



GERALDO COEN



micromega

DIVISÃO MICRODIGITAL

TK 2000/II

ENTENDENDO A ROM

Um estudo completo da ROM
de um computador
baseado no 6502

Geraldo Coen

Geraldo Coen

É um matemático que, desde 1967 trabalha em Processamento de Dados.

Atuou nas áreas de metodologia, suporte e software de base em equipamentos de médio e grande porte. Realizou interpretadores e sistemas em tempo real.

A partir de 1976, dedicou-se ao desenvolvimento de projetos de microcomputadores, tendo projetado alguns dos produtos mais difundidos no mercado nacional.

Atualmente é consultor independente em projetos de software e sistemas.

Advertência:

Esta obra é resultado de um trabalho do autor, pessoal e independente, de pesquisa da ROM do TK 2000 e não constitui uma listagem oficial ou aprovada pelo fabricante.

Apesar do esforço neste sentido e das inúmeras revisões que foram feitas, não se certifica a absoluta precisão das informações aqui contidas nem do uso que se venha a fazer delas.

Os Editores

Índice

Prefácio	5
Introdução	7
Definições.....	9
I - O Conteúdo da ROM	11
1. As Rotinas da ROM	11
2. A Distribuição dos Módulos	11
II - O Uso da RAM.....	13
1. A Distribuição das Áreas da RAM	13
III - A Operação do Interpretador	15
1. A Entrada de Linhas e Sua Análise.....	15
2. A Interpretação e Execução	16
3. As Rotinas de Execução de Comandos	17
4. As Rotinas Aritméticas e Funções	17
4.1. Formatos de Variáveis.....	17
5. As Rotinas de I/O	22
5.1. Rotina de Varredura da Teclado.....	22
5.2. Rotinas de Interpretação do Buffer IN.....	22
5.3. Saída de Dados Pelo Vídeo	23
6. Os Utilitários	28
IV - Pontos de Entrada de Rotinas	29
V - As Tabelas.....	35
Listagem da Rom	37
Bibliografia	210

Prefácio

Desde o seu lançamento, o TK 2000 tem comprovado na prática ser um computador versátil. Por estas e outras características, o TK 2000 rapidamente se tornou o mais popular computador baseado no 6502.

Seus usuários, já bastante numerosos, sentiam a necessidade de conhecer as entranhas deste computador, para poder tirar o máxima proveito em seus programas em linguagem de máquina.

Com este livro, milhares de usuários poderão satisfazer sua curiosidade a respeito de importantes detalhes do comportamento do equipamento que possuem, evoluindo mais um passo no completo domínio do TK 2000 e de um dos microprocessadores mais utilizadas no mundo toda, o 6502.

O autor teve a preocupação de não transformar este livro em um tratado técnico-científico, mas de fornecer ao usuário sua coleção de notas de trabalho, compiladas de forma prática.

Estas notas são fruto de um árduo trabalho de um especialista de software de renome, o Sr. Geraldo Coen, e serão entregues "de bandeja" aos aficionados de computação que desejavam desenvolver bons programas em linguagem de máquina.

É uma grande satisfação ver surgir uma obra desta envergadura, verdadeiramente nacional, o que representa um fato inédito na literatura técnica brasileira.

Parabéns a Geraldo Coen pela sua contribuição à tecnologia nacional de software, através da sua disposição em por em livro o fruto de seu trabalho pessoal na análise minuciosa do TK 2000.

Os Editores

Introdução

Este é um livro de software dirigida aos usuários do TK 2000 e de outras equipamentos baseadas no microprocessador 6502, e a pessoas simplesmente interessadas na matéria. Resultado de uma análise minuciosa da ROM do TK 2000, o programa, nela contido, foi estudado e comentado, em detalhes, usando minha própria experiência profissional e de bibliografia sobre a matéria, em particular no que se refere aos fundamentos de desenvolvimento de software básico, teoria de interpretadores, algoritmos numéricos.

Com a publicação completa desta análise, é meu objetivo propiciar ao leitor uma melhor compreensão de algumas técnicas importantes de software, ao mostrar um programa longo e não trivial, que inclui, entre outras a implementação de um monitor, um miniassembler, um interpretador BASIC, rotinas de comunicação com o operador, e, por última, rotinas básicas de entrada e saída (drives de I/O).

Àqueles que quiserem alterar a software básica de guia para compreensão da listagem detalhada da ROM, incluindo também a distribuição dos módulos da firmware, sua estrutura básica, além de um esquema de funcionamento e uso que é feito da RAM.

Advertimos o leitor que, para o entendimento da listagem é pressuposto o conhecimento do microprocessador 6502 e das rotinas básicas do TK 2000.

Agradeça à minha irmã Lorette, pela incentivo, e à minha esposa Gladis pelo apoio neste trabalho.

DEFINIÇÕES

Para a exata compreensão das rotinas da ROM, descritas na listagem, o leitor deverá ter conhecimento das definições dos seguintes termos:

DESCRITOR - cadeia de caracteres que descreve uma função

DP - ponto decimal (Decimal Point)

FP - ponto flutuante (Floating Point)

LITERAL - sequência de caracteres, por ex., [[caractere]]

MICROLINHA - linha de vídeo de alta resolução

NASCII - código ASCII negativos (\$80 a \$FF)

PARSE - análise da gramática; forma pela qual o computador interpreta o que está sendo digitado e coloca em ordem de execução

PASCII - código ASCII positivos (\$00 a \$7F)

SINTAXE - refere-se à estrutura dos comandos do computador, à sua ordem e às várias partes destes comandos.

STRINGS - literal fechada por aspas, por ex., "[[caractere]]"

TOKEN - código de palavras-chave referentes aos comandos da linguagem

I - O CONTEÚDO DA ROM

1. AS ROTINAS DA ROM

A ROM incorpora os seguintes grupos de rotinas:

- . Rotinas de I/O físico ("drivers")
- . Implementação do interpretador BASIC
- . Rotinas de execução dos operadores aritméticos e funções BASIC
- . Rotinas de apoio ao BASIC: alocação de memória para variáveis e tabelas, acesso a variáveis e tabelas, acesso a periféricos e outras
- . Monitor do teclado, comunicação com o operador
- . Rotinas utilitárias: LIST, LOAD etc.
- . Rotinas de inicialização: RESET etc.
- . Monitor
- . Mini-assembler

2. A DISTRIBUIÇÃO DOS MÓDULOS

A distribuição dos módulos da ROM está esquematizada no mapa a seguir (a distribuição detalhada deve ser acompanhada na listagem, pois muitas rotinas são comuns a módulos diferentes):

Mini-assembler	
Comando SOUND	
Comando PDL	
Tabelas:	
Endereços de funções	
Palavras-chave	
Mensagens	
Leitura de linhas	
Emissão de mensagens	
Análise de linhas	
Conversão de tokens	
Comandos: GOTO	
LIST	
FOR-NEXT	
STOP	
GOSUB ...	
Impressão de strings	
Avaliação de fórmulas:	
Obter variáveis	
Obter elementos de arrays	
Funções unárias	
Comparação	
Multiplicação inteira	
Execução de funções FN	
Obter espaço para strings	
Recuperar espaço livre	
Funções para strings:	
LEFT\$, ...	
PEEK, POKE	
Aritmética ponto flutuante:	
Soma, Subtração	
Multiplicação, Divisão, Log	
Movimentação de números	
Arredondamento	
Normalização	
Comparação	
Conversão em string	
SQR, EXP, RND	
SIN, COS, TAN, ATN	
Rotinas gráficas:	
Baixa resolução	
Cálculo de coordenadas	
HLINE, VLINE, PLOT, ...	
Manipulação de formas	
DRAW, XDRAW, ...	
Rotinas de I/O	

II - O USO DA RAM

1. A DISTRIBUIÇÃO DAS ÁREAS DA RAM

Durante a execução normal de programas BASIC, o TK 2000 utiliza áreas bem definidas da RAM. O mapa à esquerda mostra a distribuição destas áreas. Deve ser usado em conjunto com o mapa físico, à direita, que fornece os endereços das áreas de texto, de alta resolução, de uso das páginas 0, 1, 2, ...

ROM
\$73 - \$74 (MENSIZ)
Area de RAM
Strings
Arrays de strings
\$6F - \$70 (FRETOP)
Area Livre
\$6D - \$6E (STREND)
Arrays
\$6B - \$6C (ARYTAB)
Variáveis simples
\$69 - 6A (VARTAB)
\$AF - \$B0 (PRGEND)
Programa
\$67 - \$6B (TXTTAB)
Uso do Sistema
Páginas 0, 1, ...

0000	USO DO SISTEMA	
07FF	2 K	
0800	RAM (8K)	
1FFF		
2000	PRIMEIRA PÁG. VIDEO 8K (RAM)	MA
3FFF		
4000		
	RAM (24 K)	
9FFF		
A000	SEGUNDA PÁG. VIDEO 8K (RAM)	MP
BFFF		
C000	I/O	
C8FF		
C100	ROM	
FFFF		

As indicações no mapa correspondem ao nome das variáveis que contêm os endereços (pointers) destas áreas. Tais variáveis são definidas e usadas na listagem da ROM. Será dado aqui o endereço destas variáveis e de algumas outras importantes para a compreensão do programa.

- Notas:
1. Todos os valores estão em hexadecimal.
 2. Os endereços ocupam 2 bytes, com o byte de baixa ordem antes do byte de alta ordem.
 3. O valor decimal destas variáveis pode ser obtido através do comando `PRINT PEEK (...) + 256 * PEEK (... + 1)`.

TXTTAB	67,68	Inicio das linhas do programa; normalmente começa com 800.
VARTAB	69,6A	Inicio da área de variáveis simples, pointers a strings e a funções. Deslocado pelo comando LOMEM.
ARYTAB	6B,6C	Inicio de arrays de variáveis e strings.
STREND	6D,6E	Fim da área de arrays de variáveis e strings.
FRETOP	6F,70	Fim da área livre, fim da área de strings.
MENSTZ	73,74	Fim da área usada pelo BASIC, inicio da área de strings.
CURLIN	75,76	Número da linha que está sendo executada (linha corrente).
OLDLIN	77,78	Número da linha interrompida por CTRL-C, STOP ou END.
OLDTXT	79,7A	Endereço do último byte da última linha executada.
DATLIN	7B,7C	Número da linha da qual os dados DATA estão sendo obtidos.
DATPTR	7D,7E	Endereço do próximo byte de DATA a ler.
INPPTR	7F,80	Aponta ao buffer de teclado durante INPUT ou ao comando DATA corrente durante READ.
PRGEND	AF,80	Fim das linhas do programa.

III - A OPERAÇÃO DO INTERPRETADOR

1. A ENTRADA DE LINHAS E SUA ANÁLISE

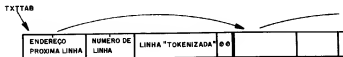
A entrada de linhas de programa obedece à seguinte seqüência:

1. Obter a próxima linha do teclado.
2. Trocar palavras reservadas por "tokens".
3. Saltar para passo 6, se for um comando de sistema (RUN ou linha sem número).
4. Guardar comando na área de programa (TXTTAB).
5. Voltar a 1.
6. Interpretar e executar.

A cada palavra-chave corresponde um "token". O token é determinado pela rotina de análise de linha PARSE. É o valor da posição da palavra-chave na tabela de palavras-chave TOKTAB mais \$80.

Exemplos: DIM (palavra) → \$80 + \$6 = \$86 (token)
HPL0T (palavra) → \$80 + \$13 = \$93 (token)

As linhas "tokenizadas" são guardadas na área que começa com TXTTAB. Cada linha está precedida de um pointer à linha seguinte e do seu número de linha, em binário. Cada linha é terminada com um byte 00.



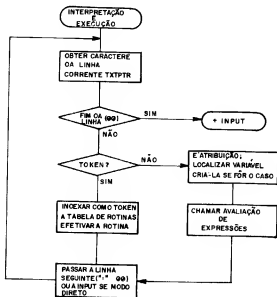
801	802	803	804	805	806	807	808
08	08	84	00	8A	41	00	
808 ENDEREÇO DO PRÓXIMO		100 EM HEXA		TOKEN PRINT	LETRA "A"	FIM DE LINHA	

EXEMPLO DA LINHA 100 PRINT A

2. A INTERPRETAÇÃO E EXECUÇÃO

A interpretação e início de execução de cada linha começa em EXECUTE. O diagrama abaixo mostra a análise do início da linha até a chamada da rotina de execução. Cada rotina de execução se encarrega de analisar o restante da linha.

No modo programado, esta rotina é executada continuamente para as linhas que estão na memória. Em modo imediato, o BASIC volta a aguardar a entrada de informações pelo teclado após esta rotina.



3. AS ROTINAS DE EXECUÇÃO DE COMANDOS

Cada rotina de execução de um comando particular analisa o restante da linha e executa o comando (INPUT, GOSUB, PRINT etc.).

Particularmente, o comando de atribuição (LET) analisa a expressão aritmética que deve ser atribuída à variável (FRNEVL).

4. AS ROTINAS ARITMÉTICAS E FUNÇÕES

Estas rotinas tratam números e strings. Os números são tratados sempre no formato interno de ponto flutuante. Observar que existem pseudo-acumuladores especiais para trabalhar com números ponto flutuante (FAC e ARG, TEMP1, TEMP2, TEMP3).

4.1. Formatos de Variáveis

4.1.1. Variáveis reais

Exemplo: A = 40000

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome 1o byte Caractere Alfabético (pos.)	41	65	A
1	da Variável 2o byte Caractere Alfanumérico (pos.)	00	0	0
2	Expoente	90	144	-
3	(M.S. byte) byte mais significativo	1C	28	-
4	Mantissa	40	64	-
5		00	0	-
6	(m.s. byte) byte menos significativo	00	0	-

Nome da variável = 1o byte deve ser um caractere alfabético

2o byte pode ser um caractere alfabético ou numérico e sua existência é facultativa

Obs.: Sendo variável real, o bit 7 do primeiro e do segundo byte deve ser 0, isto é, positivo.

O nome da variável pode conter mais do que dois caracteres, mas apenas o primeiro e o segundo são significativos, por exemplo, AB e AB345 são ambas a mesma variável.

Expoente = dado no 3o byte da variável, com valor máximo de ± 37

Obs.: 0 bit 7 deste byte dá o sinal do expoente: + (se = 0) ou - (se = 1).

Mantissa = valor da variável

No exemplo, A = 40.000
nome da variável valor da variável

4.1.2. Variáveis inteiras

Exemplo: BX = 32767

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome 1o byte Caractere Alfabético (neg.)	C2	194	B
1	da Variável 2o byte Caractere Alfanumérico (neg.)	80	128	@
2	Valor byte mais significativo	7F	127	-
3	da Variável byte menos significativo	FF	255	-
4	não usado	00	0	-
5	não usado	00	0	-
6	não usado	00	0	-

Nome da variável = 1o byte deve ser uma letra

2o byte pode ser uma letra ou um número e pode não existir

Obs.: Sendo variável inteira, o bit 7 do primeiro e do segundo byte deve ser 1, ou seja, negativo.

Valor da variável = dado pelo 3o e 4o bytes e sendo, no máximo, igual a 7FFFH (32767 em decimal)

No exemplo, BX = 32767

nome da variável valor da variável

Obs.: O byte mais significativo contém em seu bit mais significativo o sinal do valor: 0 (positivo) ou 1 (negativo).

4.1.3. Variável pointer a string

Exemplo: CD\$ = "HELLO"

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome 1o byte Caractere Alfabético (pos.) da	43	67	C
1	Variável 2o byte Caractere Alfanumérico (neg.)	C4	196	D
2	Tamanho do String	05	5	-
3	Endereço byte menos significativo do endereço do	FB	251	-
4	Parâmetro byte mais significativo do endereço	BF	143	-
5	não usado	00	0	0
6	não usado	00	0	0

Nome da variável = 1o byte deve ser um caractere alfabético

2o byte pode ser qualquer caractere alfanumérico e pode inexistir

Obs.: Sendo variável pointer a string, o bit 7 do primeiro byte será 0 (positivo), e o do segundo byte, 1 (negativo).

Tamanho do string = dá o tamanho da variável

No exemplo, "HELLO" tem cinco letras, portanto, seu valor será 5:

CD\$ = "HELLO"
12345

nome da	tamanho
variável	do string

Endereço do parâmetro = dá o endereço de memória onde se encontra o parâmetro

No exemplo, é 0FFB

Obs.: Os strings são guardados na área FRETOP, que é alocada do fim para o começo. A área dos strings livres é recuperada, quando necessário, pela rotina GARBAG, que recoloca os strings em uso de modo a eliminar espaços vagos.

4.1.4. Variáveis funções (Descritor)

Exemplo: DEF FN EF(GH) = XY

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome 1o byte Caractere Alfabético (neg.)	C5	197	E
1	Varíavel 2o byte Caractere Alfanumérico (pos.)	46	70	F
2	Endereço byte menos significativo do endereço	0E	14	-
3	Definição byte mais significativo do endereço	08	8	-
4	Endereço byte menos significativo do endereço	1B	27	-
5	Parâmetro byte mais significativo do endereço	08	8	-
6	Representação em ASCII do carac. seguinte "-"	5B	91	X

Nome da variável = 1o byte deve ser sempre uma letra

2o byte pode ser uma letra ou um número

Obs.: Sendo variável função, o bit 7 do primeiro byte será 1 (negativo), e o do segundo byte, 0 (positivo).

Endereço da definição = dá o endereço onde está a definição

No exemplo, é FN

Endereço do parâmetro = dá o endereço do parâmetro da definição

No exemplo, é (GH)

Representação em ASCII = dá o próximo caractere após o sinal de = (igual) em ASCII

No exemplo, é X

DEF	FN	EF	(GH)	=	XY
	tipo de função	nome da variável	parâmetro		valor

4.1.5. Variáveis arrays

Exemplo: DIM JK% (2,3)

BYTE	DESCRIÇÃO	HEX	DEC	ASC
0	Nome da 1o byte Caractere Alfabético (neg.)	CA	202	J
1	Variável 2o byte Caractere Alfanumérico (neg.)	C8	200	K
2	Deslocamento byte menos sign. do deslocamento da próxima	21	33	-
3	Variável byte mais signif. do deslocamento	00	0	-
4	Número de Dimensões	02	2	-
5	Tamanho da Última byte mais signif. da última dimensão	00	0	-
6	Dimensão byte menos sign. da última dimensão	04	4	-
7	Tamanho da Primeira byte mais signif. da primeira dimen.	00	0	-
8	Dimensão byte menos sign. da primeira dim.	03	3	-

nome da variável = 1o byte deve ser sempre uma letra

2o byte poderá ser uma letra ou um número

Obs.: Sendo variáveis arrays, o bit 7 do primeiro e do segundo bytes será 1 (negativo).

Deslocamento da próxima variável = dado pelo 3o e 4o bytes

Obs.: No exemplo, a matriz de 3X4 dá doze posições: (0,0); (1,0); (2,0); ... ; (2,3). Como cada posição ocupa 2 bytes, 12 (posições) X 2 = 24 bytes. Ou 24 bytes + 9 bytes da variável são iguais a 33 bytes (= 21 em hexadecimal).

Número de dimensões = número de linhas e de colunas

No exemplo, são duas (2 e 3)

Tamanho da última dimensão = 4 (0, 1, 2, 3), no exemplo

Tamanho da primeira dimensão = 3 (0, 1, 2), no exemplo

5. AS ROTINAS DE I/O

As rotinas de I/O são rotinas básicas pelas quais se processa a comunicação usuário-computador. Homem e máquina têm como canais de expressão, respectivamente, teclado e vídeo.

5.1. Rotina de Varredura do Teclado

A rotina SCAN1, alocada em \$F043, encarrega-se de fazer a varredura do teclado, verificando se há teclas acionadas.

Sendo detectada tecla(s) pressionada(s), SCAN1 verifica a posição física da matriz e, então, codifica em ASCII. O código é obtido na tabela TABASC, alocada em \$F59D.

A tabela TABASC é dividida em grupos de quatro elementos. O primeiro elemento corresponde à tecla; o segundo, à tecla em conjunto com CONTROL; o terceiro, à tecla em conjunto com SHIFT; e o quarto, à tecla em conjunto com SHIFT e CONTROL. Para cada tecla pressionada, haverá a impressão de um destes elementos no vídeo ou, então, algum flag de controle será alterado.

5.2. Rotinas de Interpretação do Buffer IN

Nas operações de entrada de dados, cada código ASCII referente à digitação é armazenado no buffer IN, até que a tecla return seja detectada. Pressionada a tecla return, o conteúdo de IN será avaliado como uma linha de informação do operador para a máquina.

Como o buffer é limitado (IN = \$0200 a \$02FF), havendo transposição do número de informações numa linha, IN reinicializa, perdendo toda a informação anterior.

Notas: 1. O TK 2000 identifica três tipos de caracteres: os alfanuméricos, os gráficos e os tokens (palavras).

2. Todos os códigos ASCII serão interpretados como alfanuméricos ou como um comando de controle.

3. A interpretação de IN inicia-se com a rotina ZXY, alocada em \$F55D.

4. Para a interpretação de IN há outro buffer, auxiliar: ING, localizado na página 6, \$0600.

Através do auxílio de ING, as rotinas principalmente interpretam o código contido em IN. Em seguida, devolvem a IN o código já interpretado, ou seja, pronto para ser trabalhado pelas outras rotinas do sistema. Isto é feito do seguinte modo:

5.2.1. Detectado um CONTROL-B (ASCII = #62), o bit menos significativo de \$0452 (GRAFFLB) será invertido, sinalizando-se a presença de um caractere gráfico.

Sendo o próximo código contido em IN um caractere ASCII, este entrará normalmente como um caractere alfanumérico.

Sendo um ASCII, antes que seja colocado em ING, um caractere de controle será posto junto (#graf = #62), e ao código ASCII será somado #540, determinando se tratar de um caractere gráfico.

Assim será até que novo CONTROL-B seja dado "desligando" a sinalização de caracteres gráficos.

Nota: Para cada caractere gráfico são usados dois bytes.

5.2.2. Não havendo um CONTROL-B, e sendo encontrado um código ASCII, a rotina irá procurar o "token" na tabela TOKTABL, alocada em \$C380. Então, os códigos ASCII correspondentes ao token serão colocados em ING, em código ASCII, até que a palavra toda seja formada.

5.3. Saída de Dados pelo Vídeo

A saída de dados pelo vídeo pode se dar por três formas:

- . Alta resolução
- . Baixa resolução
- . Texto (alfanumérico e gráfico)

A área de vídeo do TK 2000 tem duas páginas fixas de 8 Kbytes cada uma, seja para alta resolução, baixa resolução ou texto. A figura a seguir mostra uma página de vídeo:

2 000	0102	0	00
2 000	0120	1	00
2 100	0145	2	01
2 100	0175	3	02
2 200	0204	4	03
2 300	0232	5	04
2 400	0260	6	05
2 500	0288	7	06
2 600	0316	8	07
2 700	0344	9	08
2 800	0372	0	09
2 900	0400	10	0A
3 000	0428	11	0B
3 100	0456	12	0C
3 200	0484	13	0D
3 300	0512	14	0E
3 400	0540	15	0F
3 500	0568	16	10
3 600	0596	17	11
3 700	0624	18	12
3 800	0652	19	13
3 900	0680	20	14
4 000	0708	21	15
4 100	0736	22	16
4 200	0764	23	17
4 300	0792	24	18
4 400	0820	25	19
4 500	0848	26	1A
4 600	0876	27	1B
4 700	0904	28	1C
4 800	0932	29	1D
4 900	0960	30	1E
5 000	0988	31	1F
5 100	1016	32	20
5 200	1044	33	21
5 300	1072	34	22
5 400	1100	35	23
5 500	1128	36	24
5 600	1156	37	25
5 700	1184	38	26
5 800	1212	39	27
5 900	1240	40	28
6 000	1268	41	29
6 100	1296	42	2A
6 200	1324	43	2B
6 300	1352	44	2C
6 400	1380	45	2D
6 500	1408	46	2E
6 600	1436	47	2F
6 700	1464	48	30
6 800	1492	49	31
6 900	1520	50	32
7 000	1548	51	33
7 100	1576	52	34
7 200	1604	53	35
7 300	1632	54	36
7 400	1660	55	37
7 500	1688	56	38
7 600	1716	57	39
7 700	1744	58	3A
7 800	1772	59	3B
7 900	1800	60	3C
8 000	1828	61	3D
8 100	1856	62	3E
8 200	1884	63	3F
8 300	1912	64	40
8 400	1940	65	41
8 500	1968	66	42
8 600	1996	67	43
8 700	2024	68	44
8 800	2052	69	45
8 900	2080	70	46
9 000	2108	71	47
9 100	2136	72	48
9 200	2164	73	49
9 300	2192	74	4A
9 400	2220	75	4B
9 500	2248	76	4C
9 600	2276	77	4D
9 700	2304	78	4E
9 800	2332	79	4F
9 900	2360	80	50
10 000	2388	81	51
10 100	2416	82	52
10 200	2444	83	53
10 300	2472	84	54
10 400	2500	85	55
10 500	2528	86	56
10 600	2556	87	57
10 700	2584	88	58
10 800	2612	89	59
10 900	2640	90	5A
11 000	2668	91	5B
11 100	2696	92	5C
11 200	2724	93	5D
11 300	2752	94	5E
11 400	2780	95	5F
11 500	2808	96	60
11 600	2836	97	61
11 700	2864	98	62
11 800	2892	99	63
11 900	2920	100	64

EQUAD CAIXA		•	400
<input type="checkbox"/>	<input type="checkbox"/>	1024	04000
<input type="checkbox"/>	<input type="checkbox"/>	2048	08000
<input type="checkbox"/>	<input type="checkbox"/>	3072	12000
<input type="checkbox"/>	<input type="checkbox"/>	4096	16000
<input type="checkbox"/>	<input type="checkbox"/>	5120	20000
<input type="checkbox"/>	<input type="checkbox"/>	6144	24000
<input type="checkbox"/>	<input type="checkbox"/>	7168	28000

A forma mais elementar do sistema é a alta resolução, onde há 192 linhas horizontais. Na baixa resolução, os blocos são criados através de software, de quatro em quatro linhas de alta resolução. E, no modo texto, os blocos de texto (linhas) são formados de oito em oito, das linhas de alta resolução.

Os modos de alta e de baixa resolução podem ser bem entendidos na própria listagem da ROM. Neste subitem pretende-se mostrar como opera o modo texto, que é o encarregado de viabilizar a comunicação do usuário com o computador.

A principal rotina de saída de vídeo é COUT, situada em \$FDED.

Conforme o tipo de dados - caracteres alfanuméricos, caracteres gráficos ou "tokens" - haverá uma saída de dados distinta. O software se incumbirá de designar a rotina apropriada para cada tipo. Tal procedimento é descrito a seguir:

5.3.1. Inicialmente o flag GRAFFLO irá sinalizar ou não a presença de um caractere gráfico. Operando-se, então, respectivamente, um desvio para a rotina de gráficos ou para o flag VCTB. VCTB vai indicar se anteriormente registrou-se ou não CONTROL-B.

5.3.2. Não havendo CONTROL-B, e sendo detectado um "token", este receberá o tratamento da rotina ISCBAS (\$F769) que o transmutará em caracteres alfanuméricos.

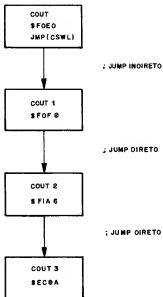
5.3.3. Havendo um CONTROL-B, será realizado um exame para detectar a presença de Mgraf (\$SF2).

Se for positiva a presença de Mgraf, a rotina setará GRAFFLO para que o próximo caractere seja tratado como um caractere gráfico.

Este processo se interromperá somente quando for detectado outro CONTROL-B, "desligando" o modo gráfico.

5.3.4. Estando os caracteres separados em todas as suas formas, a rotina IDFFCOD (\$FBFD) desempenhará a ação das teclas de funções de vídeo. E, ainda, a rotina NXTPOS1 (\$FBF0) buscará um bloco de oito bytes para cada caractere, em GERCHR (\$F200) ou em GERGRF, conforme os caracteres sejam alfanuméricos ou gráficos, posicionando-os na página de vídeo referente à posição onde cada caractere é desejado.

Os fluxogramas abaixo permitem uma visualização das rotinas descritas neste subitem:



Seqüência de prioridade das rotinas de saída de dados

6. OS UTILITÁRIOS

As rotinas utilitárias tratam comandos como CLEAR, DELETE, NEW, RUN, LIST etc.

São comandos executados diretamente. Após sua execução, o BASIC volta a solicitar a introdução de dados pelo teclado.

IV - PONTOS DE ENTRADA DE ROTINAS

Destacam-se aqui os principais pontos de entrada:

CHRGCT	B1	carrega A de TXTPTR, incrementando-o.
CHRGOT	B7	carrega A de TXTPTR.
LINGET	CD31	converte um número de linha de TXTPTR a LINUM.
GTBYT	DA12	avalia a fórmula em TXTPTR, deixando o resultado em FAC e passando a CONINT.
PLOTFNS	E540	obtem 2 coordenadas baixa resolução de TXTPTR.
HFNS	EA3A	obtem 2 coordenadas alta resolução de TXTPTR.
FMULT	DC99	move o número de Y,A a ARG e passa a FMULTT.
FMULTT	DC9C	multiplica FAC por ARG em FAC.
FDIV	DD80	move o número de Y,A a ARG e passa a FDIVT.
FDIVT	DD83	divide ARG por FAC, resultado em FAC.
FADD	DAD8	move o número de Y,A a ARG e passa a FADDT.
FADDT	DAD8	soma FAC com ARG em FAC.
FSUB	DAC1	move o número de Y,A a ARG e passa a FSUBT.
FSUBT	DAC4	subtrai FAC de ARG, resultado em FAC.
FPWRT	E181	eleva ARG à potência FAC.
SNG	DEAA	chama SIGN e "flutua" o resultado em FAC.
ABS	DEC9	valor absoluto de FAC.
INT	DF3D	maior valor inteiro de FAC. Usa @INT e "flutua" o resultado.
SQR	E1A7	raiz quadrada de FAC.
LOG	DC58	logaritmo base e de FAC.
EXP	E223	eleva e à potência FAC.
RND	E2CB	pseudonôm

TAN	E354	tangente de FAC.
ATN	E3B8	arco tangente de FAC.
MOVFM	DE13	move de Y,A a FAC.
MOV2F	DE38	compacta FAC e move-o a TEMP2.
MOV1F	DE3B	compacta FAC e move-o a TEMP1.
MOVHL	DE3D	compacta FAC e move-o à área apontada por X.
MOVHF	DE45	compacta FAC e move-o a Y,X.
MOVFA	DE6D	move ARG a FAC.
MOVAF	DE7D	move FAC a ARG.
CONUPK	DCFD	carrega ARG de Y,A.
SIGN	DE9C	coloca em A 1, 0 ou FF, conforme FAC seja positivo, zero ou negativo.
FOUT	E04E	monta em FBUFFER um string equivalente ao conteúdo de FAC. Destrói FAC. Imprime o string com STROUT.
FCOMP	DECC	compara FAC com o número compactado em Y,A. Coloca em A 1, 0 ou FF, conforme Y,A seja (, = ou >) que FAC.
NEGOP	E1EA	faz $FAC = - FAC$.
FADDDH	DABA	faz $FAC = FAC + 1/2$.
DIV10	D06F	divide FAC por 10.
MUL10	D053	multiplica FAC por 10.
SNGFLT	D618	"flutua" o inteiro em Y.
GIVAYF	D60C	"flutua" o inteiro com sinal em A,Y.
FLOAT	DEAD	"flutua" o inteiro com sinal em A.
CONINT	DA15	converte FAC em número de um só byte em X e FACLO.
AYINT	D426	se FAC menor que 32767 e maior que - 32767, chama @INT.
@INT	DF0C	função maior inteiro.
GETADR	DA6C	converte o número de FAC em número binário de 2 bytes em LINNUM.

GETNUM	FFAA	obtem um número de 2 bytes em LINNUM de TXTPTR, passa uma vírgula e obtém um número de 1 byte em X.
COMBYTE	DA66	passa uma vírgula e obtém 1 byte em X.
FRMEVL	D6A3	avalia em FAC a fórmula de TXTPTR.
FRMNUM	D08FF	avalia em FAC a fórmula de TXTPTR. Verifica se é um número.
FIN	DF64	entrada de número ponto flutuante em FAC.
CAT	D8B1	concatena 2 strings.
STRINI	D6EF	obtem espaço para um string e monta seu descritor.
COPY	CDDC	libera o string apontado por Y,A e move-o à memória apontada por FORPNT.
MOVINS	D8EE	move o string cujo descritor é apontado por STRNG1 à memória apontada por FRESPA.
MOVSTR	D8FC	move o string apontado por Y,A à memória apontada por FRESPA.
STRLIT	D701	põe uma aspa em ENDOCHR e CHARAC.
STRLT2	D707	monta um descritor para o literal de Y,A.
GETSPA	D76C	obtem espaço para string. Pode forçar uma recuperação de espaço livre.
FRETMP	D91E	libera um string temporário.
FRETHS	D94F	libera um descritor de string.
INLIN	C81B	recebe uma linha de texto.
INCHR	C86B	obtem um caractere do teclado.
STROUT	CE5F	mostra no vídeo o string de Y,A.
STRPRT	CE62	mostra o string com o descritor dado.
OUTDO	CEB1	mostra o caractere de A.
CRDO	CE20	realiza um CR (carriage return).
INPRT	E033	mostra "IN" e o número de linha de CURLIN.
LINPRT	E03E	mostra o número de 2 bytes de X,A.
PRNTFAC	E04B	mostra o valor de FAC. Destrói FAC.

PTRGET	D303	obtem um nome de variável e localiza-o na memória. TXTPTR aponta ao primeiro caractere do nome. Calcula o endereço da variável em VARPNT. Se não encontrar a variável, ela é criada.
GETARYPT	EB59	o mesmo que PTRGET para array. Não cria variável se não a encontrar.
FNDLIN	C931	procura no programa a linha de LINNUM. Coloca seu endereço em LOWTR.
DATA	CCBA	desloca TXTPTR para o fim da linha corrente.
DATAN	CCC8	calcula em Y o deslocamento de TXTPTR ao próximo fim de linha "s" ou "ø".
REMN	CCC8	calcula em Y o deslocamento de TXTPTR ao próximo fim de linha "ø".
ADDON	CCBD	soma Y a TXTPTR.
SCRTCH	C962	efetua o comando NEW.
CLEARC	C9B3	efetua o comando CLEAR.
STKTNI	C99A	limpa a pilha.
RESTORE	CB69	coloca em DATPTR, pointer de DATA, o endereço de início do programa.
STXPT	C9AE	coloca em TXTPTR o endereço de início do programa.
BLTU	C6B2	move um bloco de bytes para a frente na memória, para criar espaço.
REASON	C6D2	verifica se há espaço na memória.
GARBAG	D79E	recupera espaço livre, eliminando strings não utilizados.
CONT	CB8B	move OLDTXT e OLDLIN a TXTPTR e CURLIN.
NEWSTT	CAEC	executa uma nova linha do programa.
RUN	CC37	executa o programa da memória.
GOTO	CC63	atualiza TXTPTR a partir de LINGET e FNDLIN.
LET	CD68	avaliação da atribuição em TXTPTR. Usa CHRGET para obter o endereço da variável, avalia a fórmula e guardá o resultado.
HRR2	E748	inicializa e limpa a página 2 de alta resolução.

HGR	E74F	inicializa e limpa a página 1 de alta resolução.
BKGND	E775	enche a tela alta resolução com a última cor usada.
HPPOSN	E790	posiciona o cursor alta resolução. A página é determinada por HPAG.
HPL0T	FA7E	"plota" um ponto na posição do cursor.
HLIN	E586	traca uma linha do último ponto ao ponto dado pelos registradores X,A (horizontal) e Y (vertical).
HFIN	E946	converte a posição do cursor alta resolução a coordenadas x,y. Usada após SHAPE para se localizar.
DRAW	EAE9	desenha a forma apontada por Y,X usando a cor HCOLOR. A = fator de rotação.
XDRAW	EAEF	desenha a forma apontada por Y,X invertendo a cor dos pontos existentes. A = fator de rotação.
SETCOL	F864	faz a cor de alta resolução igual a X.
SHLOAD	EAF5	carrega a tabela de formas de fita em MENSIZ e prepara o pointer em \$E8.
SAVE	CB05	grava o programa da mendria na fita.
LOAD	CBEE	carrega programa a partir de fita.
ERROR	C701	imprime mensagem de erro ou desvia a rotina de erro caso ONERR ativo.
HANDLERR	E64A	prepara variáveis em função do erro cujo código está em X.
RESUME	E679	volta após erro, restaura variáveis.
ISCNTC	CB78	verifica se foi teclado CTRL-C. Se foi, chama BREAK.
CHKNUM	D092	verifica se FAC é numérico.
CHKSTR	D094	verifica se FAC é string.
CHKVAL	D095	verifica se o resultado da última operação em FAC foi string ou número, para conferir compatibilidade de tipos de variáveis.

ISLETC	D397	verifica se A contém uma letra ASCII. C é ligado se A contém uma letra.
PARCHK	D1DA	verifica a existência de "(", avalia a fórmula e verifica a existência de ")".
CHKCLS	D1E0	verifica a existência de ")" em TXTPTR.
CHKOPN	D1E3	verifica a existência de "(" em TXTPTR.
CHKCOM	D1E6	verifica a existência de "," em TXTPTR.
SYNCHR	D1EB	verifica se o caractere em TXTPTR é igual ao caractere em A.

V - AS TABELAS

Encontram-se na ROM as seguintes tabelas:

CMDTABL	C2D6	endereços das rotinas de execução de cada comando e função.
MATHBL	C392	endereços das rotinas de operações aritméticas, precedidos por um valor hexadecimal que corresponde a sua prioridade numa expressão.
TOKTABL	C380	tabela de nomes dos comandos para cálculo do seu "token".
MENSERR0	C54F	tabela de mensagens de erro.
GERCHR	F200	tabela de geração de caracteres de texto.
GERGRF	F400	tabela de geração de caracteres gráficos.
TABASC	F59D	códigos ASCII referentes ao teclado.
CHRTBL	FFCA	funções do monitor.
FMT1	F962	
FMT2	F9A6	
CHAR1	F9B4	tabelas referentes aos comandos do miniassembler, disassembler.
CHAR2	F9BA	
MNEML	F9C0	
MNEHR	FA00	

LISTAGEN DA ROM


```

1  #####
2  *
3  *      ROM  TR2000
4  *
5  *      Listagem comentada
6  *
7  *      De:
8  *
9  *      $C100 a $FFFF
10 *
11 *
12 *
13 #####
14 *
15 *      Mini-assembler
16 *
17 *      Comando SOUND,POL
18 *
19 *      Entrada parsing
20 *
21 *      Rotinas de endereçamento
22 *
23 *      FOR-NEXT loops,etc
24 *
25 #####
26
27 *Localizacao da pag. zero
28
29 GOWARM EQU $00      ;Preparado por cold start
30 LOCD EQU $00        mas nao usado por outros
31 LOC1 EQU $01        ;LOCs,locacoes de memoria
32 LOC2 EQU $02        usadas pelo MONITOR
33 LOC3 EQU $03
34 GOSTROUT EQU $03
35 LOC4 EQU $04
36 R3L EQU $06         ;REL-H,pseudo registradores
37 R3H EQU $07         de 16 bits
38 R4L EQU $08
39 R4H EQU $09
40 UR EQU $0A
41 CHARAC EQU $00      ;Usados para strings
42 ENDCR EQU $0E
43 MROTH EQU $0F       ;Usado em rotinas de
44 PTR EQU $0F         manipulacao de arrays
45 OIPFL EQU $10
46 VALTYP EQU $11      ;$FF para string,0 se numerico
47 INTFL EQU $12       ;Idem para variavel inteira
48 DATAFL EQU $13     ;Usado no PARSE
49 GANFL EQU DATAFL   ;Usado em GANING
50 SUBFL EQU $14       ;Flag de subscritos
51 INPUTFL EQU $15     ;$NOGET, $PORGAD
52 CPANASK EQU $16     ;Recebe CPRTYP em FINEVL
53 SIGMFL EQU $16     ;Indica sinal para TAN
54 SHAPFL EQU $1A      ;Ponteiros usados pelas
55 SHAPEN EQU $1B      rotinas de HI-RES
56 NCOLOR1 EQU $1C     ;Usado pela rotina HI-RES
57 COUNTN EQU $1D     ;Usado pela rotina HI-RES
58 MODLFT EQU $20      ;Ponteiros de formatacao
59 MDMOTH EQU $21      de quadro de video

```


60	WHDTP	EMU	\$22	
61	WHDTH	EMU	\$23	
62	CH	EMU	\$24	;Coordenadas de cursor
63	CV	EMU	\$25	
64	GBASL	EMU	\$26	;Ponteiros de linha
65	GBASH	EMU	\$27	;da rotina PLOT
66	BASL	EMU	\$28	;Ponteiros de linha
67	BASH	EMU	\$29	de texto
68	BASZL	EMU	\$2A	;Dados durante scrolling
69	BASZH	EMU	\$2B	
70	LINEH	EMU	\$2C	;Usado pelo disassembler
71	H2	EMU	\$2C	;Usado por DRAW e VLINE
72	V2	EMU	\$2D	;Usado por DRAW e VLINE
73	RNDH	EMU	\$2E	;Usado pelo disassembler
74	FORMAT	EMU	\$2E	;Mini-assembler disassembler
75	CHSUM	EMU	\$2E	;Ponteiros das rotinas
76	LASTIH	EMU	\$2F	;de cassette
77	LENGTH	EMU	\$2F	;Usado pelo disassembler
78	MMASK	EMU	\$30	;Mascara da rotina HI-RES
79	COLOR	EMU	\$30	;Conteúdo a ser plotado
80	MODE	EMU	\$31	;Usado pelo monitor
81	INPFLG	EMU	\$32	;Formato normal ou inverso
82	PROMPT	EMU	\$33	;Contém o caractere de pronto
83	YSAV	EMU	\$34	;Salva conteúdo de Y
84	ENDMASK	EMU	\$35	
85	YSAVI	EMU	\$35	
86	CSML	EMU	\$36	;Ponteiro da rotina de OUT
87	KSM	EMU	\$38	;Ponteiro da rotina de IN
88	PCL	EMU	\$3A	;Programa counter RES.
89	PCH	EMU	\$3B	
90	AIL	EMU	\$3C	;AOL-H, registradores de
91	AIH	EMU	\$3D	uso geral
92	AOL	EMU	\$3E	
93	AZH	EMU	\$3F	
94	AOL	EMU	\$40	
95	AZH	EMU	\$41	
96	AAL	EMU	\$42	
97	AZH	EMU	\$43	
98	ASL	EMU	\$44	
99	ACC	EMU	\$45	;Salva acumulador
100	AAG	EMU	\$46	
101	PBVL	EMU	\$47	;Ponteiro do endereço
102	PBWH	EMU	\$48	da página de vídeo
103	VCTB	EMU	\$49	;FLAG de caractere gráfico
104	BNDL	EMU	\$4E	
105	BNDH	EMU	\$4F	
106	LTHAM	EMU	\$50	;Uso geral, número de linha
107	TEMPPT	EMU	\$52	;Ponteiros temporários
108	LASTPT	EMU	\$53	
109	TEMPST	EMU	\$55	
110	INDEX	EMU	\$5E	
111	DEST	EMU	\$60	;Parâmetros de operações
112	RESULT	EMU	\$62	matemáticas
113	XTTIN	EMU	\$67	;De \$67 até \$80, ver
114	BASLOC	EMU	\$67	II-0 USO DA RAM
115	VARTIN	EMU	\$69	
116	ARTIN	EMU	\$6B	

117	STREND	END	\$6D	
118	FRETOP	END	\$6F	
119	FRESPEC	END	\$71	
120	MENSIZ	END	\$73	
121	CURLIN	END	\$75	
122	QURLIN	END	\$77	
123	QLOTEXT	END	\$79	
124	DATLIN	END	\$7B	
125	DATPTR	END	\$7D	
126	IMPTR	END	\$7F	
127	WARHAM	END	\$81	; \$10-,\$11-,\$realize, jumpo-+
128	WARPNT	END	\$83	; Ponteiro de variavel
129	FORPNT	END	\$85	; Uso geral
130	TOPSV	END	\$87	; Usado por INPUT
131	LASTOP	END	\$87	; Ind.de trab.usado em FINEV.
132	CPRTOP	END	\$89	; Ind.)=(em FINEV.
133	FIRCHAM	END	\$8A	
134	TEMP3	END	\$8A	; REG. de ponto flutuante
135	OSCPTR	END	\$8C	
136	OSCLEH	END	\$8F	; Usado em GABRAG
137	JMPADRS	END	\$90	; Ponteiro de instrcao JMP
138	ALCMTIN	END	\$91	; Usado em GABRAG
139	EXTRASV	END	\$92	; Mais preciso para F.P.
140	TEMP1	END	\$93	; Areas de trabalho para FAC
141	HIGHDS	END	\$94	; Usado em GABRAG
142	ARIPNT	END	\$94	
143	HIGHTR	END	\$96	; Usado para transferir bloco
144	TEMP2	END	\$96	; REG. de ponto flutuante
145	INDX	END	\$99	; Usado pelas rotinas de arrays
146	TEMPX	END	\$99	; Usado em FINEVAL
147	EXPON	END	\$9A	; Usado em FINEVAL
148	DIFLG	END	\$9B	; Indica ponto decimal em FIN
149	LOWTR	END	\$9B	; Uso geral
150	EXPSON	END	\$9C	; Indica sinal do expoente em FIN
151	FAC	END	\$9D	; p/um.ponto flutuante principal
152	OSCTMP	END	\$9D	
153	VVNT	END	\$A0	; #pointer a variavel temporaria
154	FACSON	END	FACVS	; Sinal descompactado
155	SERLEN	END	\$A3	; Tamanho da serie - 1
156	FPGEN	END	\$A4	; Uso geral em FP
157	ARG	END	\$A5	; acumulador pto flut.second.
158	ARGSON	END	ARGVS	
159	SGNCPD	END	\$A8	; Ind.sinal de oper.em rot. F.P.
160	STRMS1	END	\$A8	; Utilizado para strings
161	EXTRAFAC	END	\$AC	; Preciso F.P.
162	SERPNT	END	\$AD	; p/ont.aos dados da serie em f.p.
163	STRMS2	END	\$AD	; Utilizado para strings
164	PROEND	END	\$AF	; Ponteiro de fim de programa
165	BASELOC	END	\$B0	; Endereco do interp. BASIC
166	CHRGST	END	\$B1	; busca de caracter
167	CHRGST	END	\$B7	
168	TXTPTR	END	\$B8	; p/ponto proximo carac/token
169	RNDSEED	END	\$C9	; numero randomico F.P
170	ODL	END	\$D0	; \$D0-\$D6, ponteiros de
171	ODI	END	\$D1	tratamento nas rotinas de erro
172	DI	END	\$D2	
173	ODONT	END	\$D3	

174	EL	EMU	\$04	
175	EN	EMU	\$05	
176	LOCK	EMU	\$06	
177	ERRFLG	EMU	\$08	;Trava acesso pelo usuario, se negativo
178	ERRLIN	EMU	\$0A	
179	ERRPOS	EMU	\$0C	
180	ERRHUM	EMU	\$0E	
181	ERRSTX	EMU	\$0F	
182	XCL	EMU	\$10	;Coordenada X em HI-RES
183	XDR	EMU	\$E1	
184	YO	EMU	\$E2	;Coordenada Y em HI-RES
185	DELH	EMU	\$E3	;Ponteiro de deletacao de linha
186	MOBLINZ	EMU	\$E4	;Usado em HI-RES
187	HMAX	EMU	\$E5	
188	HPAG	EMU	\$E6	;Aponta a pagina HI-RES
189	SCALEZ	EMU	\$E7	;Fator de escala em HI-RES
190	SHAPEMT	EMU	\$E8	
191	CALCOUNT	EMU	\$EA	
192	KEMTC	EMU	\$EF	
193	FIRST	EMU	\$FD	;Primeira coordenada de PLOT
194	SPEEDZ	EMU	\$F1	;Velocidade de Output
195	TROFLG	EMU	\$F2	
196	ORMASK	EMU	\$F3	;Mascara para controle de OUT
197	TXTPSN	EMU	\$F4	
198	CMLSV	EMU	\$F6	
199	RENSTX	EMU	\$F8	
200	ROTZ	EMU	\$F9	
201	COOL	EMU	\$FB	;Uso geral do monitor
202	COOH	EMU	\$FC	
203	RTMP1	EMU	\$FD	;REG. temporario do monitor
204				
205	STACK	EMU	\$100	;Pagina 1
206				
207	IN	EMU	\$0200	;Pagina 2, buffer de teclado
208				
209	BREVL	EMU	\$03FD	;Pagina 3, vetor do BREAK
210	SOFTVEA	EMU	\$03F2	;Vetor de entrada do BASIC
211	SOFTVEH	EMU	\$03F3	
212	PMKEDUP	EMU	\$03F4	
213	AMPER	EMU	\$03F5	;Comparacao do RESET
214	TK2030	EMU	\$03FB	
215	LBRA04	EMU	\$03FC	
216	MHI	EMU	\$03FB	
217	IRNLACL	EMU	\$03FE	
218				
219	PGA	EMU	\$0403	;Pagina4
220	TSUTIL	EMU	\$0405	;Teste de utilitarios
221	FLGRAM	EMU	\$0440	;Teste de memoria
222	TEST41	EMU	\$0441	
223	TEST42	EMU	\$0442	
224	TEST50	EMU	\$0450	
225	TEST51	EMU	\$0451	
226	BRAFFLO	EMU	\$0452	;Indica se graficos
227	TEST53	EMU	\$0453	
228	BUF	EMU	\$0460	
229	PGR1	EMU	\$0468	
230	PGR2	EMU	\$046A	

229	TEST70	EMU	\$0470	
232	TEST71	EMU	\$0471	
233	TEST72	EMU	\$0472	
234	TEST73	EMU	\$0473	
235	TEST74	EMU	\$0474	
236	FINLIN	EMU	\$0482	;Fin de linha
237	TEST86	EMU	\$0486	
238	TEST80	EMU	\$0480	
239	XSAY	EMU	\$048E	;Salva REG. X
240				
241	IMG	EMU	\$0600	;Pagina 6, buffer auxiliar
242				
243	AREG	EMU	\$07F0	;Pagina 7
244	XREG	EMU	\$07F1	
245	YREG	EMU	\$07F2	
246	STATUS	EMU	\$07F3	
247	SPAT	EMU	\$07F4	
248	RLAX	EMU	\$07FF	
249				
250	PRG1	EMU	\$0800	;Pagina 8
251	PRG3	EMU	\$0803	
252	PRG6	EMU	\$0806	
253	PRG9	EMU	\$0809	
254	PRGC	EMU	\$080C	
255	PRGF	EMU	\$080F	
256	PRG12	EMU	\$0812	
257	PRG15	EMU	\$0815	
258	PRG18	EMU	\$0818	
259				
260	ESPR1AK	EMU	\$47FF	
261	R48K	EMU	\$87FF	
262	CART0	EMU	\$9000	
263	CART3	EMU	\$9003	
264	CART6	EMU	\$9006	
265	CART9	EMU	\$9009	
266	NRFL01	EMU	\$9854	
267	NRFL62	EMU	\$9855	
268	TSCART	EMU	\$9FF0	;Testa cartucho
269	INICART	EMU	\$9FF1	;Inicia cartucho
270	KROUT	EMU	\$C000	;Saída de dados pelo teclado
271	KBTPTN	EMU	\$C010	;Entrada de dados pelo teclado
272	TAPEOUT	EMU	\$C020	;Saída de dados para gravador
273	SPKH	EMU	\$C030	;Saída de som
274	DECOLOR	EMU	\$C0C0	;Imagem colorida
275	ISPER	EMU	\$C051	;Imagem preto e branco
276	NOTAOFF	EMU	\$C052	;Comando NOTCH
277	DISPAG1	EMU	\$C054	;Primeira pagina de video
278	DISPAG2	EMU	\$C055	;Segunda pagina de video
279	PRTSTH	EMU	\$C059	;Printer strobe high
280	BTCONT0	EMU	\$C05E	;Bit control em nível 0
281	BTCONT1	EMU	\$C05F	;Bit control em nível 1
282	POLO	EMU	\$C064	;Sinal de Padie
283	INICPOL	EMU	\$C070	;Inicializacao
284	PRF1	EMU	\$C080	
285	PRF2	EMU	\$C088	
286	OSK	EMU	\$C100	
287	INPR	EMU	\$C200	

```

288 ENDMIMP EQU $807A ;Endereco nao importante
289
290 *Tx2000soft tokens:
291
292 far EQU $B1
293 data EQU $B3
294 bell EQU $B7
295 pop EQU $A1
296 goto EQU $A8
297 gosub EQU $B0
298 ren EQU $B2
299 print EQU $BA
300 tab EQU $C3
301 to EQU $C1
302 fa EQU $C2
303 spc EQU $C5
304 lven EQU $C4
305 at EQU $C5
306 not EQU $C6
307 step EQU $C7
308 plus EQU $C8
309 minus EQU $C9
310 equal EQU $D0
311 sgn EQU $D2
312 scrn EQU $D7
313 leftstr EQU $E8
314 graf EQU $F2
315
316 PUT PARTELA,D1
>1 ORG $C100
>2
>3 *MINI-ASSEMBLER
>4
C100: E9 B1 >5 RELAT SRC #B01 ;Calcula deslocamento relativo
C102: 4A >6 LSR
C103: 00 54 >7 BNE RELEMB
C105: A4 3F >8 LDI A2H ;Endereco destino
C107: A6 3E >9 LOX A2L
C109: 00 01 >10 BNE RELAT1
C10B: 00 >11 DEX
C10C: CA >12 RELAT1 DCX
C10D: 8A >13 TXA
C10E: 18 >14 CLC
C10F: E5 3A >15 SRC PCL
C111: 05 3E >16 STA A2L ;Armazena o deslocamento
C113: 10 01 >17 BPL RELAT2
C115: C8 >18 DRY
C116: F0 >19 RELAT2 TYA
C117: E5 30 >20 SRC PCN
C119: 00 40 >21 RELEMB BNE INDEMB1 ;Erro,excede +127,-126 BYTES
C11B: A4 2F >22 TANDIST LDI LENGTH ;D=1,1=2,2=3 (BYTES)
C11D: 09 3D 00 >23 BUSHYT LDM AUX,Y
C120: 91 3A >24 FORKDIST STA (PCL),Y ;armaz.no end.indic.per PC+LENGTH
C122: 00 >25 DEX
C123: 10 F0 >26 BPL BUSHYT
C125: 20 23 FC >27 JBR UP ;Sube cursor de 2 posicoes
C126: 20 23 FC >28 JBR UP

```

C120:	20 00 F8 129	JSR	INSTOSP	;Disassembla instrucao no video	
C12C:	20 53 F9 130	JSR	PCADJ	;Ajusta novo valor do PC	
C131:	04 38 131	STR	PCB		
C139:	05 3A 132	STA	PCD		
C13F:	4C 95 C1 133	JMP	CONTASS	;Continua assembling	
C13B:	20 86 EE 134	WFCOM	JSR	TOSUB1	;Verifica comandos (NA,HP,PA,MA)
C13B:	AA 34 135	LDI	YSBV	;Busca posicao do buffer IN	
C130:	20 AA FF 136	ORTEND	JSR	GETMAN	;Obtem endereco hexadecimal
C140:	04 34 137	STY	YSBV	;Salva posicao do buffer IN	
C142:	A0 17 138	LDI	MS17	;ultima posicao de CHRTBL	
C144:	08 139	PROGCOM	DEY		
C145:	30 48 140	BRI	ASS	;Nao e nenhum, reinicia em ASS	
C147:	09 CA FF 141	CHP	CHRTBL, Y	;E algum comando de CHRTBL ?	
C14A:	00 F8 142	BNE	PROCOM	;Nao, proximo comando ate o fim	
C14C:	C0 15 143	CPY	MS15	;E RETURN ?	
C14E:	00 D8 144	BNE	WFCOM		
C150:	A5 31 145	LDA	MODE		
C152:	A0 00 146	LDI	MS00		
C154:	C6 34 147	DEC	YSBV	; Volta posicao no buffer IN	
C15A:	20 D0 FE 148	JSR	BL1	;Subst. ou impr. contido de mem.	
C159:	4C 95 C1 149	JMP	CONTASS	;Busca nova linha p/ assembler	
C15C:	A5 30 150	PROCESS	LDA	AUX	;assembla OPC00
C15E:	20 91 F8 151	JSR	MS053		
C161:	AA 152	TAX			
C162:	80 00 FA 153	LDA	WHEN, X		
C165:	C5 42 154	CMF	AAL		
C167:	00 13 155	BNE	PROCESS1		
C169:	80 C0 F9 156	LDA	WHEN, X		
C16C:	C5 43 157	CMF	AAR		
C1AE:	00 DC 158	BNE	PROCESS1		
C170:	A5 44 159	LDA	ASL		
C172:	A4 2E 160	LDI	FORMAT		
C174:	C0 90 161	CPY	MS00	;E enderecamento relativo ?	
C176:	F0 08 162	BEQ	RELAT	;Sim, RELAT	
C178:	C5 2E 163	CMF	FORMAT		
C17A:	F0 9F 164	BEQ	TIMEMST		
C17C:	C6 30 165	PROCESS1	DEC	ASH	
C17E:	00 DC 166	BNE	PROCESS		
C180:	E6 44 167	INC	ASL		
C182:	C6 35 168	DEC	ENHNGR		
C184:	F0 D6 169	BEQ	PROCESS		
C186:	AA 34 170	INDEX1	LDI	YSBV	;Recupera posicao do buffer
C188:	90 171	INDEXR	TYA		
C189:	AA 172	TAX			
C18A:	20 4A F9 173	JSR	PROBL2	;Imprime X espacos	
C18D:	A5 DE 174	LDA	***		
C18F:	20 ED FD 175	JSR	OUT	;Imprime ***	
C192:	20 3A FF 176	ASS	JSR	BELL	;Pto de entrada do miniassembler
C195:	A9 A1 177	CONTASS	LDA	B"1"	
C197:	85 33 178	STA	PROMPT		
C199:	20 67 FD 179	JSR	RETLR2	;Obtem linha de entrada de dados	
C19C:	28 EF EE 180	JSR	ZY00	;Zera MODE	
C19F:	A0 D0 D2 181	LDA	IN	;E inicio do buffer IN	
C1A2:	C9 A0 182	CMF	B" "	;E linha para disassemblar ?	
C1A4:	F0 13 183	BEQ	ASSEMB	;Sim	
C1A6:	C8 184	INY			
C1A7:	C9 AA 185	CMF	B"5"		

CJAF: F0 F2	186	BER	ONTEND	;Sim
CJAG: 00	187	DEY		
CJAC: 20 AA FF	188	JSR	GETNUM	;Obtem endereço hexa em A2
CJAF: C9 F3	189	CMF	#F03	;Compara com "1"
CJAG: 00 05	190	INVALID	BNE INDEXR	;Indica posicao do erro
CJG3: 0A	191	TXA		
CJAG: F0 02	192	BER	INDEXR	
CJAG: 20 78 FE	193	JSR	ALFCLP	;Transfere endereço de A1 para PC
CJAF: A9 03	194	ASSEMB	LDA #F03	;valida OPC00
CJAG: 05 30	195		STA A1H	
CJAG: 20 34 C2	196	ASSEMB1	JSR BUSESP	;ignora espacos
CJGD: 0A	197	ASL		
CJCL: E9 BE	198	SBC	#B6E	
CJG3: C9 C2	199	CMF	#C2	;OPC00 valido ?
CJCS: 90 C1	100	BCC	INDEXR	;Mao erro
CJG7: 0A	101	ASL		
CJCB: 0A	102	ASL		
CJG9: A2 04	103	LDX	#04	
CJCB: 0A	104	ASSEMB2	ASL	;Memorico em A4
CJCC: 26 42	105	ROL	AAL	
CJCE: 26 43	106	ROL	AHH	
CJG0: CA	107	DEX		
CJG1: 10 F0	108	BPL	ASSEMB2	
CJG3: CA 30	109	DEC	A1H	
CJG5: F0 F4	110	BER	ASSEMB2	
CJG7: 10 E4	111	BPL	ASSEMB1	
CJG9: A2 05	112	LOX	#A05	;Ultima posicao de CHM1,CHM2
CJG0: 20 34 C2	113	CONTECH	JSR BUSESP	;Ignora espacos no buffer
CJG1: 04 34	114	TIPEND	STY YSMV	;Salva Y
CJG2: 00 84 F9	115	CMF	CHM1,X	;Verifica forma de enderecamento
CJG3: 00 03	116	BNE	TIPEND1	
CJG5: 20 34 C2	117	JSR	BUSESP	;Ignora espacos no buffer
CJG0: 00 8A F9	118	CMF	CHM2,X	
CJG1: F0 00	119	BER	IMMEDIAT?	;Se CHM1 e CHM2
CJG2: 00 8A F9	120	LDA	CHM2,X	
CJF0: F0 07	121	BER	TIPEND2	
CJF2: C9 A4	122	CMF	#"5"	;E "5" ?
CJF4: F0 03	123	BER	TIPEND2	
CJF6: AA 34	124	LOI	YSMV	
CJF8: 18	125	TIPEND1	CLC	
CJF9: 00	126	TIPEND2	DEY	
CJFA: 26 44	127	IMMEDIAT?	BOL ASL	
CJFC: E0 03	128	CPX	#03	;CHM1=0 , CHM2=5
CJFE: 00 00	129	BNE	IMMEDIAT	;Mao e immediato?
CJG0: 20 AA FF	130	JSR	GETNUM	
CJG3: A5 3F	131	LDA	A2H	
CJG5: F0 01	132	BER	ZPGW	;Se pagina zero
CJG7: E8	133	IMX		;Se absoluto
CJG0: 06 35	134	ZPGW	STX ENDMASK	
CJG4: A2 03	135	LDX	#03	
CJG6: 00	136	DEY		
CJG9: 06 30	137	IMMEDIAT	STX A1H	
CJG1: CA	138	DEX		
CJG3: 10 C9	139	BPL	CONTECH	
CJG5: A5 44	140	LDA	ASL	
CJG1: 0A	141	ASL		
CJG3: 0A	142	ASL		

C244:	85 35	1143	ORA	ENDMASK	
C248:	C7 20	1144	CMP	#520	
C25A:	80 04	1145	BCC	FINFEND	;Se >=20
C26C:	A4 35	1146	LDX	ENDMASK	
C21E:	F0 02	1147	BEQ	FINFEND	;Se =0
C228:	87 80	1148	ORA	#580	;Se > 0 e < 20, 000000
C222:	85 44	1149	FINFEND	STA	ASL
C224:	84 34	1150	STY	YSAV	;Salva posicao do buffer IN
C226:	87 00 02	1151	LDA	IN,Y	
C229:	C7 88	1152	CMP	#1"	;E "1" ?
C22B:	F0 04	1153	BEQ	JPROCESS	;Sim
C22D:	C7 80	1154	CMP	#600	;E RETURN ?
C22F:	80 80	1155	BNE	INVALID	;Nao, linha invalida
C231:	AC 5C C1	1156	JPROCESS	JMP	PROCESS
C234:	87 00 02	1157	BUSESP	LDA	IN,Y
C237:	C8	1158	INY		;Assabla linha
C238:	C7 A0	1159	CMP	#"	;E "espaco" ?
C23A:	F0 FB	1160	BEQ	BUSESP	;Sim, entao novo caracter
		1161			
C23C:	A0	1162	RTS		;Nao, retorne
C23D:	FF FF FF	1163	HEX	FFFFFF	
		1164			
C240:	20 12 DA	1165	SOUND	JSR	GETBYT
C243:	86 40	1166	STX	ASL	;Obtem a nota
C245:	A7 2C	1167	LDA	#", "	;Guarda em ASL
C247:	20 E8 01	1168	JSR	SYNCH	;Compara com ", "
C24A:	20 12 DA	1169	JSR	GETBYT	;Obtem a duracao da nota
C24D:	86 41	1170	STX	ASH	;Guarda em ASH
C24F:	A0 39 C0	1171	TOWAL	LDA	SPKR
C252:	88	1172	DURACAO	DEY	;Prod. um "click" no alto-falante
C253:	00 05	1173	BNE	FREQ	
C255:	CE 41 00	1174	DEC:	ASH	
C258:	F0 09	1175	BEQ	FINNOTA	;Acabou a nota ?
C25A:	CA	1176	FREQ	DEX	
C25B:	80 F5	1177	BNE	DURACAO	
C25D:	AE 40 00	1178	LOX:	ASL	
C260:	AC 4F C2	1179	JMP	TOWAL	
C263:	20 87 00	1180	FINNOTA	JSR	CHRGOT
C266:	C7 C1	1181	CMP	#0	;Tem mais nota ?
C268:	F0 01	1182	BEQ	PROXNOT	
C26A:	60	1183	RTS		
C26B:	20 81 00	1184	PROXNOT	JSR	CHRGOT
C26E:	AC 40 C2	1185	JMP	SOUND	
		1186			
		1187			*Botina reservada de paddle, para uma possivel interface
		1188			na ser desenvolvida
		1189			*POLO=SC06A, PDL1=SC065, PDL2=SC066, PDL3=SC067
		1190			
C271:	20 15 DA	1191	POL	JSR	CONJMT
C274:	A0 70 C0	1192	LDA	INICPOL	;Inicia paddle
C277:	A0 00	1193	LDT	#500	
C279:	EA	1194	MOP		;Oria atrazo
C27A:	EA	1195	MOP		
C27B:	80 64 C0	1196	SENEG	LDA	POLO,X
C27E:	10 DA	1197	BPL	FINTEMP	;Negativo ate terminar o tempo
C280:	C8	1198	INY		
C281:	80 F8	1199	BNE	SENEG	


```

C283: 80      1200      DET
C284: 4C 18 06 1201  FINTENP  JNP 88F8LT
      1202
C287: 80      1203      DFB 880
C288: 80      1204      DFB 880
C289: 80      1205      DFB 880
C28A: 82      1206      DFB 882
C28B: C8      1207      DFB 8C8
C28C: 04      1208      DFB 8D4
C28D: 0A      1209      DFB 8EA
C28E: 20      1210      DFB 820
C28F: F5      1211      DFB 8F5
C290: F8      1212      DFB 8F8
C291: 20      1213      DFB 820
C292: 73      1214      DFB 873
C293: FC      1215      DFB 8FC
C294: A5      1216      DFB 8A5
C295: 08      1217      DFB 808
C296: 85      1218      DFB 885
C297: 80      1219      DFB 880
C298: A5      1220      DFB 8A5
C299: 09      1221      DFB 809
C29A: 85      1222      DFB 885
C29B: 81      1223      DFB 881
C29C: 18      1224      DFB 818
C29D: A5      1225      DFB 8A5
C29E: 84      1226      DFB 884
C29F: 0A      1227      DFB 80A
C2A0: 0A      1228      DFB 80A
C2A1: A5      1229      DFB 8A5
C2A2: 81      1230      DFB 881
C2A3: 85      1231      DFB 885
C2A4: 81      1232      DFB 881
C2A5: A0      1233      DFB 8A0
C2A6: FF      1234      DFB 8FF
C2A7: FF      1235      DFB 8FF
      1236
C2A8: A5 47    1237  TESTCAR  LDA 88A      ;Testa acessórios
C2A9: F0 00    1238      BEB 80A8
C2AC: 4C 00 08 1239      JNP 8801
C2AF: A0 35 04 1240  80A8  LDA 8801LT
C2B0: C9 02    1241      CNP 8802
C2B4: 30 08    1242      BRL 8808      ;Nao tem ?
C2B6: C9 02    1243      CNP 8802
C2B8: F0 0F    1244      BEB 880F
C2BA: C9 03    1245      CNP 8803
C2BC: 00 04    1246      BNE 8804
C2BE: 80 46 00 1247  SETCFLG  STA 88A4
C2C1: A0      1248  FINTENP  RTS
C2C2: C9 04    1249  SET4     CNP 8804
C2C4: 00 03    1250      BNE 880F
C2C6: 4C 8E C2 1251      JNP SETCFLG
C2C9: 4C 04 00 1252  880F  JNP (LOCK)
      1253
C2CC: FF FF FF 1254      HEX FFFFFFFF
C2CF: FF
      1255

```

C200:	4C	42	E4	1256	BASIC	JMP	COLIST	;Pontos de entrada do BASIC
				1257				
C203:	4C	28	C7	1258	BASIC2	JMP	RESTART	
				317		PUT	PARTE18,01	
				31				
C204:	94	08		12	CROTABL	DA	END-1	
C209:	7F	0A		13		DA	FOR-1	
C20A:	20	00		14		DA	NEXT-1	
C20C:	89	0C		15		DA	DATA-1	
C20E:	06	0E		16		DA	INPUT-1	
C210:	91	06		17		DA	DEL-1	
C212:	F8	02		18		DA	DIA-1	
C214:	04	0F		19		DA	READ-1	
C216:	F8	0A		110		DA	ER-1	
C218:	02	E7		111		DA	TEXT-1	
C21A:	FF	00		112		DA	DSK-1	
C21C:	91	C1		113		DA	ASS-1	
C21E:	35	E5		114		DA	CALL-1	
C21F:	78	E5		115		DA	PLOT-1	
C221:	85	E5		116		DA	HLIN-1	
C224:	94	E5		117		DA	VLIN-1	
C226:	47	E7		118		DA	HGR2-1	
C228:	5A	E7		119		DA	HGR-1	
C22A:	69	EA		120		DA	HCOL,00-1	
C22C:	70	EA		121		DA	HPLOT-1	
C22E:	6B	EA		122		DA	ORAM-1	
C230:	EE	EA		123		DA	XDRAM-1	
C232:	66	EB		124		DA	HTAB-1	
C234:	57	FC		125		DA	HOME-1	
C236:	40	EA		126		DA	ROT-1	
C238:	A6	EA		127		DA	SCALE-1	
C23A:	F4	EA		128		DA	SHL,00-1	
C23C:	03	E5		129		DA	TRACE-1	
C23E:	05	E5		130		DA	NOTRACE-1	
C240:	09	E5		131		DA	NORMAL-1	
C242:	0C	E5		132		DA	INVERSE-1	
C244:	3F	C2		133		DA	SOUND-1	
C246:	A2	E5		134		DA	COLOR-1	
C248:	8F	CC		135		DA	POP-1	
C24A:	8C	E5		136		DA	VTAB-1	
C24C:	6A	E5		137		DA	HOPEN-1	
C24E:	06	EA		138		DA	LOPEN-1	
C250:	28	EA		139		DA	OPEN-1	
C252:	78	EA		140		DA	RESUME-1	
C254:	28	E7		141		DA	RECALL-1	
C256:	0E	E7		142		DA	STORE-1	
C258:	C8	E5		143		DA	SPEED-1	
C25A:	4A	C0		144		DA	LET-1	
C25C:	62	CC		145		DA	GOTO-1	
C25E:	36	CC		146		DA	RUN-1	
C260:	ED	CC		147		DA	IF-1	
C262:	68	C8		148		DA	RESTORE-1	
C264:	F4	03		149		DA	APPEND-1	
C266:	45	CC		150		DA	GOSUB-1	
C268:	8F	CC		151		DA	POP-1	;Ponto de entrada de RETURN
C26A:	00	C0		152		DA	REN-1	
C26C:	92	C8		153		DA	STOP-1	

C30E: 10 C0	154	DA	ONBOTO-1	
C340: 70 DA	155	DA	WATTO-1	
C342: 91 E8	156	DA	RLDAD-1	
C344: C3 E8	157	DA	RSNVE-1	
C346: 2C D6	158	DA	DEF-1	
C348: 94 DA	159	DA	POKE-1	
C34A: F9 C0	160	DA	PRDRT-1	
C34C: 8A C8	161	DA	CONT-1	
C34E: 88 C9	162	DA	LIST-1	
C350: 90 C9	163	DA	CLEAN-1	
C352: C4 CE	164	DA	GET-1	
C354: 5F C9	165	DA	MEM-1	
C356: AA DE	166	DA	SEN	
C358: 30 DF	167	DA	INT	
C35A: C7 DE	168	DA	NBS	
C35C: DA 00	169	DA	USR	
C35E: F8 05	170	DA	FRE	
C360: D1 C7	171	DA	ERROR	Ponto de entrada da rotina SCIN
C362: 70 C2	172	DA	POL-1	
C364: 19 D6	173	DA	POS	
C366: A7 E1	174	DA	SAR	
C368: C8 E2	175	DA	RWD	
C36A: 58 DC	176	DA	LOG	
C36C: 23 E2	177	DA	EXP	
C36E: 04 E3	178	DA	COS	
C370: 08 E3	179	DA	SIN	
C372: 54 E3	180	DA	TAN	
C374: 88 E3	181	DA	ATH	
C376: 7C DA	182	DA	PDEK	
C378: F0 D9	183	DA	LEN	
C37A: DF D6	184	DA	STR	
C37C: 21 DA	185	DA	VAL	
C37E: FF D9	186	DA	ASC	
C380: 60 D9	187	DA	CHRSTR	
C382: 74 D9	188	DA	LEFTSTR	
C384: 40 D9	189	DA	RIGHTSTR	
C386: AA D9	190	DA	NDSTR	
C388: 60 FF	191	DA	NAME-1	; (LX)
C38A: 7C E8	192	DA	NOTOR-1	
C38C: F7 D3	193	DA	TX2000-1	
C38E: F7 E8	194	DA	HP-1	
C390: FF E8	195	DA	NA-1	
	196			
	197			* Os 16 hex sao usados no teste de prioridade*
	198			
C392: 79	199	DATA	HEX	79
C393: DA DA	1100	DA	FADDT-1	; +
C395: 79	1101	HEX	79	
C396: C3 DA	1102	DA	FSUBT-1	; -
C398: 78	1103	HEX	78	
C399: 98 DC	1104	DA	FNULTT-1	; *
C39B: 78	1105	HEX	78	
C39C: 82 D0	1106	DA	FDIVT-1	; /
C39E: 70	1107	HEX	70	
C39F: 80 E1	1108	DA	FPART-1	; ^
C3A1: 50	1109	HEX	50	
C3A2: 7C D2	1110	DA	AND-1	; AND

C3A4: 46	1111	HEX	46	
C3A5: 76 02	1112	DA	06-1	; OR
C3A7: 7F	1113	KIDUS	HEX	7F
C3A8: E7 E1	1114	DA	NEGOP-1	;Menos usuario
C3AA: 7F	1115	UNIT	HEX	7F
C3AB: BF D1	1116	DA	ENOP-1	;NOT usuario
C3AD: 64	1117	PLUS	HEX	64
C3AE: 0C 02	1118	DA	POSOP-1	;Usado por (-)
	1119			
C3B0: 45 4E C4	1120	TOKTABL	DC1	'END'
C3B3: 46 4F 02	1121		DC1	'FOR'
C3B4: 4E 45 58	1122		DC1	'NEXT'
C3B9: 04				
C3BA: 44 41 54	1123		DC1	'DATA'
C3B4: C1				
C3BE: 49 4E 50	1124		DC1	'INPUT'
C3C1: 55 04				
C3C3: 44 45 0C	1125		DC1	'DEL'
C3C6: 44 49 C9	1126		DC1	'DIR'
C3C7: 52 45 41	1127		DC1	'READ'
C3CC: C4				
C3CD: 47 02	1128		DC1	'OR'
C3CF: 54 45 58	1129		DC1	'TEXT'
C3D2: 04				
C3D3: 44 53 C8	1130		DC1	'DIR'
C3D4: 41 53 03	1131		DC1	'ASS'
C3D9: 43 41 4C	1132		DC1	'CALL'
C3DC: CC				
C3DE: 50 4C 4F	1133		DC1	'PLOT'
C3DD: 04				
C3E1: 48 4C 49	1134		DC1	'LUN'
C3E4: CE				
C3E5: 56 4C 49	1135		DC1	'LUN'
C3E8: CE				
C3F1: 48 47 52	1136		DC1	'RGR2'
C3EC: 02				
C3ED: 48 47 52	1137		DC1	'RGR'
C3FD: 46 43 4F	1138		DC1	'HCOLON'
C3F3: 4C 4F 02				
C3FA: 48 50 4C	1139		DC1	'PLOT'
C3FF: 4F 04				
C3FB: 44 52 41	1140		DC1	'DRAW'
C3FE: 07				
C3FF: 58 44 52	1141		DC1	'EDRAW'
C402: 41 07				
C404: 48 54 41	1142		DC1	'HTAB'
C407: C2				
C408: 48 4F 40	1143		DC1	'HOME'
C40B: C3				
C40C: 52 4F 54	1144		DC1	'ROT='
C40F: 80				
C410: 53 43 41	1145		DC1	'SCALE='
C413: 4C 45 80				
C416: 53 48 4C	1146		DC1	'SHLWD'
C419: 4F 41 C4				
C41C: 54 52 41	1147		DC1	'TRACE'
C41F: 43 C3				

C421: 4E 4F 54 1148	DCI 'NOTRACE'
C424: 52 41 43 C5	
C428: 4E 4F 52 1149	DCI 'NORMAL'
C42B: 40 41 CC	
C42E: 49 4E 54 1150	DCI 'INVERSE'
C431: 45 52 53 C5	
C435: 53 4F 55 1151	DCI 'SOUND'
C438: 4E C4	
C43A: 43 4F 4C 1152	DCI 'COLOR'
C43B: 4F D2	
C43F: 50 4F D0 1153	DCI 'POP'
C442: 54 54 41 1154	DCI 'VTAB'
C445: C2	
C44A: 48 49 40 1155	DCI 'HOMER1'
C449: 45 4D BA	
C44C: 4C 4F 40 1156	DCI 'LOWER1'
C44F: 45 4D BA	
C452: 4F 4E 45 1157	DCI 'ONERR'
C455: 52 D2	
C457: 52 45 53 1158	DCI 'RESUME'
C45A: 55 4D C5	
C45D: 52 45 43 1159	DCI 'RECALL'
C460: 41 4C CC	
C463: 53 54 4F 1160	DCI 'STORE'
C466: 52 C5	
C46B: 53 50 45 1161	DCI 'SPEED='
C46B: 45 44 B0	
C46E: 4C 45 D4 1162	DCI 'LET'
C471: 47 4F 54 1163	DCI 'GOTO'
C474: C7	
C475: 52 55 CE 1164	DCI 'RUN'
C47B: 49 C6 1165	DCI 'IF'
C47A: 52 45 53 1166	DCI 'RESTORE'
C47D: 54 4F 52 C5	
C481: A6 1167	ASC 'A'
C482: 47 4F 53 1168	DCI 'EOLIN'
C485: 55 C2	
C487: 52 45 54 1169	DCI 'RETURN'
C48A: 55 52 CE	
C48B: 52 45 CD 1170	DCI 'REN'
C490: 53 54 4F 1171	DCI 'STOP'
C493: D0	
C494: 4F CE 1172	DCI 'ON'
C496: 57 41 49 1173	DCI 'WAIT'
C499: D4	
C49A: 4C 4F 41 1174	DCI 'LOAD'
C49D: C4	
C49E: 53 41 56 1175	DCI 'SAVE'
C4A1: C5	
C4A2: 44 45 C6 1176	DCI 'DEF'
C4A5: 5D 4F 48 1177	DCI 'PONE'
C4A8: C5	
C4A9: 5D 52 49 1178	DCI 'PRINT'
C4AC: 4E D4	
C4AE: 43 4F 4E 1179	DCI 'CONT'
C4B1: D4	
C4B2: 4C 49 53 1180	DCI 'LIST'

CAB5: 04	
CAB6: 43 4C 43 1181	DCI 'CLEAN'
CAB7: 41 02	
CAB8: 47 45 04 1182	DCI 'GET'
CABE: 4E 45 07 1183	DCI 'NEW'
CAC1: 54 41 42 1184	DCI 'TAM1'
CAC4: A8	
CAC5: 54 0F 1185	DCI 'TO'
CAC7: 46 0E 1186	DCI 'FN'
CAC9: 53 50 43 1187	DCI 'SPC1'
CACC: A8	
CACD: 54 48 45 1188	DCI 'THEN'
CAD0: CE	
CAD1: 41 04 1189	DCI 'AT'
CAD3: 4E 4F 04 1190	DCI 'NOT'
CAD6: 53 54 45 1191	DCI 'STEP'
CAD9: D0	
CAD8: A8 1192	ASC '4'
CAD8: A0 1193	ASC '5'
CADC: AA 1194	ASC '6'
CAD0: AF 1195	ASC '7'
CAD6: DE 1196	ASC '8'
CAD7: 41 4E C4 1197	DCI 'AND'
CAD2: 4F 02 1198	DCI 'OR'
CAD4: BE 1199	ASC '9'
CAD5: B0 1200	ASC '0'
CAD6: BC 1201	ASC '1'
CAD7: 53 47 0E 1202	DCI 'SON'
CAD8: 4F 4E 04 1203	DCI 'INT'
CAD0: 41 42 03 1204	DCI 'AND'
CAD0: 53 53 02 1205	DCI 'USR'
CAD3: 46 52 C5 1206	DCI 'FRE'
CAD4: 53 43 52 1207	DCI 'SCRM'
CAD9: 4E A8	
CAD8: 50 44 CC 1208	DCI 'POL'
CAD5: 50 4F 03 1209	DCI 'POS'
CS01: 53 51 02 1210	DCI 'SON'
CS04: 52 4E C4 1211	DCI 'RND'
CS07: 4C AF C7 1212	DCI 'LOG'
CS0A: 45 50 00 1213	DCI 'EXP'
CS00: 43 4F 03 1214	DCI 'COS'
CS10: 53 49 CE 1215	DCI 'SIN'
CS13: 54 41 CE 1216	DCI 'TAN'
CS16: 41 54 CE 1217	DCI 'MTH'
CS19: 50 45 45 1218	DCI 'PEEX'
CS1C: D8	
CS1D: 4C 45 CE 1219	DCI 'LEN'
CS20: 53 54 52 1220	DCI 'STR5'
CS23: A4	
CS24: 5A 41 CC 1221	DCI 'VAL'
CS27: 41 53 C3 1222	DCI 'ASC'
CS2A: 43 48 52 1223	DCI 'OW5'
CS20: A4	
CS2E: 4C 45 46 1224	DCI 'LEFT5'
CS31: 54 A4	
CS33: 52 49 47 1225	DCI 'RIGHT5'
CS36: 48 54 A4	

```

CS39: 40 49 44 1225      DCI 'WDS'
CS3C: 4A
CS3D: 4C 0D 1227      DCI 'LH'
CS3F: 40 4F 54 1228      DCI 'NOTON'
CS42: 4F 02
CS44: 54 4B 32 1229      DCI 'TK2000'
CS47: 30 30 80
CS4A: 40 00 1230      DCI 'HP'
CS4C: 40 C1 1231      DCI 'MA'
CS4E: 00 1232      HEX 00
          368      PUT PARTEIC,01
          3E
          32      NOXERR
CS4F: 4E 45 58 13      NOXERR DCI 'NEXT SEN FOR'
CS52: 54 20 53 45 40 20 46 4F
CS5A: 02
CS5B: 53 49 4E 34      SENTERRO DCI 'SINTAXE '
CS5E: 54 41 58 45 A0
CS63: 52 45 54 15      RTNERR DCI 'RETURN SEN GOSUB'
CS64: 53 52 4E 20 53 45 40 20
CS6E: 47 4F 53 55 C2
CS73: 4E 41 4F 36      NHAData DCI 'NAO HA MAIS DATA'
CS76: 20 4B 41 20 40 41 49 53
CS7E: 20 44 41 54 C1
CS83: 56 41 4C 37      VALILEG DCI 'VALOR ILEGAL'
CS86: 4F 52 20 49 4C 45 47 41
CS8E: CC
CS8F: 45 53 54 38      ESTOR DCI 'ESTOURO'
CS92: 4F 55 52 CF
CS96: 45 58 43 19      EXCERR DCI 'EXCEDE MEMORIA'
CS99: 45 44 45 20 40 45 40 4F
CSA1: 52 49 C1
CSA4: 43 4F 40 310      COMDEF DCI 'COMANDO NAO DEFINIDO'
CSA7: 41 4E 44 4F 20 4E 41 4F
CSAF: 20 14 45 46 49 4E 49 44
CSB7: CF
CSB8: 49 4E 44 311      INDILEG DCI 'INDICE ILEGAL'
CSB8: 49 43 45 20 49 4C 45 47
CSCB: 41 CC
CSC5: 52 45 27 312      REDIMARR DCI 'RE'DIR' DE ARRAYS'
CSCE: 44 49 40 27 29 44 45 20
CSCE: 41 52 52 41 09
CSD5: 44 49 56 313      DIVZERO DCI 'DIVISAO POR ZERO'
CSD8: 49 53 41 4F 20 50 4F 52
CSE9: 20 56 45 52 CF
CSE5: 43 4F 40 314      COMILER DCI 'COMANDO ILEGAL'
CSER: 41 4E 44 4F 20 49 4C 45
CSF0: 47 41 CC
CSF3: 49 4E 43 315      INCOMP DCI 'INCOMPATIVEL'
CSFA: 4F 40 50 41 54 49 56 45
CSFE: CC
CSFF: 45 58 43 316      EXCSTR DCI 'EXCEDE STRING'
GAB2: 45 44 45 20 53 54 52 49
GADA: 4E C7
GADC: 46 4F 52 317      FORMuCX DCI 'FORMULA MUITO COMPLEXA'
GAFP: 40 55 4C 41 20 40 55 49
GAB7: 54 4F 20 43 4F 40 50 4C

```

```

C63F: 45 50 C1
C640: 49 40 50 >18  INPSSL DCI 'IMPOSSIVEL'
C641: 4F 53 53 49 54 45 CC
C642: 46 55 4E >19  FUNvDEF DCI 'FUNCAO NAO DEFINIDA'
C643: 43 41 4F 20 4E 41 4F 20
C644: 44 45 46 49 4E 49 44 C1
      >20
C64F: 20 23 45 >21  ERRIN  ASC 'ERRO'0700
C650: 52 52 4F 07 00
C647: 20 45 40 >22  INMSG  ASC 'EM '00
C646: 20 00
C64C: 00 >23  BREAKIN HEX 00
C64B: 42 52 45 >24  ASC 'BREAK'0700
C65B: 41 4B 07 00
      >25
C654: 8A >26  GTFORPNT TSX ;Procurar na pilha
C655: EB >27  JMX
C656: EB >28  JMX
C657: EB >29  JMX
C65B: EB >30  JMX
C659: 80 01 01 >31  FNEFOR  LDA STACK+1,X
C65C: C7 81 >32  CMP #40r
C65E: 00 21 >33  BNE RET1
C660: A5 86 >34  LDA FORPNT+1
C662: 00 0A >35  BNE SAMEFOR? ;Se variavel especificada
C664: 80 02 01 >36  LDA STACK+2,X ;Obter pointer a variavel FOR
C667: 85 85 >37  STA FORPNT
C669: 80 03 01 >38  LDA STACK+3,X
C66C: 85 86 >39  STA FORPNT+1
C66E: 00 03 01 >40  SAMEFOR? CMP STACK+3,X ;Comp enderecos de variavel FOR
C671: 00 07 >41  BNE NZFOR ;Diferentes,desviar
C673: A5 85 >42  LDA FORPNT
C675: 00 02 01 >43  CMP STACK+2,X
C678: F0 07 >44  BEQ RET1
C67A: 8A >45  NZFOR  TXA ;FOR incorreto
C67B: 1B >46  CLC ;Preparar para pesq.o proximo
C67C: 6F 12 >47  ADC #642
C67E: AA >48  TAX
C67F: 00 00 >49  BNE FNEFOR
C681: 60 >50  RET1  RTS
      >51
C682: 20 02 C6 >52  BLTU  JSR REASON ;Ha espaco?
C685: 85 60 >53  STA STREND ;Fim da area de array=(A,T)
C687: 84 4E >54  STY STREND+1
      >55
      ; Preparar para mover de LONTR ate HIGHTR-1
      ; para a area a partir do byte seguinte a HIGHDS
      >56
      >57
      >58
C689: 3B >59  BLTU2  SEC
C68A: A5 96 >60  LDA HIGHTR
C68C: E3 9B >61  SBC LONTR
C68E: 85 9E >62  STA INDEX
C690: A0 >63  TAY
C691: A5 97 >64  LDA HIGHTR+1
C693: E3 9C >65  SBC LONTR+1
C695: AA >66  TAX
C696: EB >67  JMX

```


C697:	98	168	TTA	
C698:	FD 23	169	BEQ	KOPAG ;Se nao pag parcial,desviar
C69A:	45 96	170	LDA	HIGHT0 ;frep,para mover pag,parcial
C69C:	38	171	SEC	;antes,para maximizar velocid.
C69D:	E3 5E	172	SBC	INDEX
C69F:	85 96	173	STA	HIGHT0
C6A1:	8D 03	174	BCS	SETEND
C6A3:	C6 97	175	DEC	HIGHT0+1
C6A5:	38	176	SEC	
C6A6:	A5 94	177	SETEND	LDA HIGH05
C6A8:	E3 5E	178	SBC	INDEX
C6AA:	85 94	179	STA	HIGH05
C6AC:	8D 08	180	BCS	WSETT
C6AE:	C6 95	181	DEC	HIGH05+1
C6B0:	9D 04	182	BCC	WSETT
C6B2:	B1 96	183	WSETT	LDA (HIGHT0),Y
C6B4:	91 94	184	STA	(HIGH05),Y
C6B6:	88	185	WSETT	BEY
C6B7:	D0 F9	186	BNE	WSETT
C6B9:	B1 96	187	LDA	(HIGHT0),Y
C6BB:	91 94	188	STA	(HIGH05),Y
C6BD:	C6 97	189	KOPAG	DEC HIGHT0+1
C6BF:	C6 95	190	DEC	HIGH05+1
C6C1:	CA	191	DEX	;Outra pagina a mover?
C6C2:	D0 F2	192	BNE	WSETT
C6C4:	6D	193	BTS	
		194		
		195		*Teste de espaco na pilha usado por F00, G0SUB, F0RML
		196		
C6C5:	DA	197	CHROMEN	ASL ;Entra com A+1,3 ou 9
C6C6:	69 36	198	ADC	W36
C6C8:	8D 35	199	BCS	WENEN0
C6CA:	85 5E	100	STA	INDEX
C6CC:	8A	101	TSX	
C6CD:	E4 5E	102	CPX	INDEX
C6CF:	9D 2E	103	BCC	WENEN0
C6D1:	6D	104	BTS	
		105		
C6D2:	C4 7D	106	BEASON	CPY FRET0P+1 ;Verifica se A,Y < FRET0P
C6D4:	9D 28	107	BCC	SET2 ;Sim,retorna
C6D6:	D0 04	108	BNE	W51 ;Nao,limpa area
C6D8:	C5 6F	109	CHP	FRET0P
C6DA:	9D 22	110	BCC	SET2
C6DC:	48	111	W51	PHA ;Guardar A,Y e TEMP1 e TEMP2
C6DE:	A2 09	112	LDX	WAC-TEMP1-1
C6DF:	98	113	TTA	
C6E0:	48	114	W52	PHA
C6E1:	85 9D	115	LDA	TEMP1,X
C6E3:	CA	116	DEX	
C6E4:	1D FA	117	BPL	W52
C6E6:	2D 9E 07	118	JSR	WABW0 ;Reorganizar a area
C6E9:	A2 F7	119	LDX	WTEMP1-FAC+1
C6EB:	68	120	W52	PLA ;Restaurar TEMP1,TEMP2 e A,Y
C6ED:	9D 9D	121	STA	FAC,X
C6EE:	E8	122	END	
C6F0:	3D FA	123	BNE	W53
C6F1:	68	124	PLA	

C6F2: 40	1125	TAY	
C6F3: 48	1126	PLA	;Agora tem espaço?
C6F4: C4 70	1127	CPY FRET0P+1	
C6F6: 90 06	1128	BCC RET2	;Sim,retornar
C6F8: 00 05	1129	BNE NEMERR	;Nao,erro falta de memoria
C6FA: C5 AF	1130	CHP FRET0P	
C6FC: 80 01	1131	BDS NEMERR	
C6FE: 60	1132	RET2 RTS	
	1133		
C6FF: A2 47	1134	NEMERR LDX MEMMEN-NEMSEIRO	
C701: 24 00	1135	ERR08 BIT ERRFLG	;ONERR ativo?
C703: 10 03	1136	BPL DOERRMSG	;Nao,desviar
C705: 4C 4A E6	1137	JMP HANDLE08	
	1138		
C708: 20 20 CE	1139	DOERRMSG JSR CR00	
C70A: 20 7F CE	1140	JSR OUTQUES	
C70E: 80 4F C5	1141	ERLUP LDA NEMSEIRO,X	
C711: 48	1142	PHA	
C712: 20 81 CE	1143	JSR OUT00	
C715: E8	1144	INX	
C716: 68	1145	PLA	
C717: 10 F5	1146	BPL ERLUP	
C719: 20 9A C9	1147	JSR STK1W1	
C71C: A9 3F	1148	LDA NERR10	
C71E: A0 C4	1149	LDY BERR08	
C720: 20 5F CE	1150	PRINTIN? JSR STROUT	
C723: A4 76	1151	LDY CURLIN+1	;Modo direto
C725: C8	1152	INY	
C726: F8 03	1153	BEQ RESTART	;Sim, desviar
C728: 20 33 E0	1154	JSR IMPRT	
	1155		
C72B: 20 20 CE	1156	RESTART JSR CR00	
C72E: A2 BE	1157	LDX B?"	
C730: 20 10 C8	1158	JSR IMLIN2	;Obter input direto
C733: 86 88	1159	STX TXTPTR	;Apontar ao buffer de input
C735: 84 89	1160	STY TXTPTR+1	
C737: A6 08	1161	LSR ERRFLG	;Ligar ONERR
C739: 20 B1 00	1162	JSR CHRGET	
C73C: A8	1163	TAX	
C73D: F8 EC	1164	BEQ RESTART	;Se nao ha input
C73F: A2 FF	1165	LDX #FF	;ligar indicador de modo direta
C741: 86 76	1166	STX CURLIN+1	;byte de alta ordem de CURLIN
C743: 90 04	1167	BCC KOL10	;Desv.se houver numer.de linha
C745: 20 6E C8	1168	JSR GET10	;Caso contrario,analizar
C748: 4C 1F C8	1169	JMP TRACE?	e executar
C74B: A6 AF	1170	KOL10 LDX PINGEND	
C74D: 86 69	1171	STX WARTAN	
C74F: A6 80	1172	LDX PINGEND+1	
C751: 86 6A	1173	STX WARTAN+1	
C753: 20 31 C8	1174	JSR LIMRET	;Obter numero da linha
C756: 20 AE C8	1175	JSR GETIN	e analisar input
C759: 84 0F	1176	STY PTR	;guardar indice no buffer input
C75B: 20 31 C9	1177	JSR FIMLIN	;ha alguma linha?
C75E: 90 44	1178	BCC REMAN?	;Nao,desviar
C760: A0 01	1179	LDY #01	;Sim,apagar a linha
C762: B1 98	1180	LDA (LONTR),Y	;Obter o apontador
C764: 85 5F	1181	STA INDEX+1	

C766: A5 6F	1182	LDA	VARTAB	
C76B: 05 5E	1183	STA	INDEX	
C76A: A5 9C	1184	LDA	LOWTR+1	
C76C: 05 41	1185	STA	DEST+1	
C76E: A5 9B	1186	LDA	LOWTR	
C770: 00	1187	DEY		
C771: F1 9B	1188	SBC	(LOWTR),Y	;Linha-apontador
C773: 18	1189	CLC		
C774: 65 6F	1190	ADC	VARTAB	
C776: 05 6F	1191	STA	VARTAB	;Novo fim de programa
C77B: 05 60	1192	STA	DEST	
C77A: A5 6A	1193	LDA	VARTAB+1	
C77C: 6F FF	1194	ADC	#FF	
C77E: 05 6A	1195	STA	VARTAB+1	
C780: E3 9C	1196	SBC	LOWTR+1	
C782: AA	1197	TAX		
C783: 30	1198	SEC		
C784: A5 9B	1199	LDA	LOWTR	
C786: E3 6F	1200	SBC	VARTAB	
C78B: A0	1201	TAY		;Índice para mover pag parcial
C789: 00 03	1202	RCS	HL1	
C78F: 00	1203	INX		
C78C: C6 61	1204	DEC	DEST+1	
C78E: 18	1205	HL1	CLC	
C78F: 65 5E	1206	ADC	INDEX	
C791: 90 03	1207	RCC	MUDAR	
C793: C6 5F	1208	DEC	INDEX+1	
C795: 18	1209	CLC		
C796: 01 5E	1210	LDA	(INDEX),Y	;Mover resto do programa
C798: 91 60	1211	STA	(DEST),Y	;ao espaço da linha apagada
C79A: C8	1212	JMY		
C79B: 00 F9	1213	BNE	MUDAR	
C79D: E6 5F	1214	INC	INDEX+1	
C79F: E6 61	1215	INC	DEST+1	
C7A1: CA	1216	DEX		;Mover outra pagina?
C7A2: 00 F2	1217	BNE	MUDAR	
C7A4: A0 00 02	1218	MEMLN?	LDA	IN
C7A7: F0 38	1219	BEQ	LINKSET	;Se numero de linha?
C7A9: A5 73	1220	LDA	MEMSIZ	;Sim,saltar a LINKSET
C7AB: A4 74	1221	LDY	MEMSIZ+1	
C7AD: 05 6F	1222	STA	FRETOP	
C7AF: 04 70	1223	STY	FRETOP+1	
C7B1: A5 6F	1224	LDA	VARTAB	;Preparar movimento de memoria
C7B3: 05 96	1225	STA	HIGHTR	para inserir nova linha
C7B5: 65 0F	1226	ADC	PWTR	
C7B7: 05 94	1227	STA	HIGHDS	
C7B9: A4 6A	1228	LDY	VARTAB+1	
C7BB: 04 97	1229	STY	HIGHTR+1	
C7BD: 90 01	1230	BCC	MUPRS	
C7BF: C8	1231	JMY		
C7C0: 04 95	1232	MUPRS	STY	HIGHDS+1
C7C2: 20 02 C6	1233	JSR	BLTU	;Realizar o movimento
C7C5: A5 50	1234	LDA	LINKMUR	
C7C7: A4 51	1235	LDY	LINKMUR+1	
C7C9: 00 FE 01	1236	STA	TR-2	
C7CC: 0C FF 01	1237	STY	IN-1	
C7CF: A5 60	1238	LDA	STREND	

C701:	A4 6E	1239	LDY	STREND+1	
C703:	B5 6F	1240	STA	WARTAB	
C705:	B4 6A	1241	STY	WARTAB+1	
C707:	A4 6F	1242	LDY	PXTR	
C709:	B9 F8 01	1243	INSRTLIN	LDA	IN-S,Y
C70C:	B0	1244	DEY		;insrerir nova linha
C70D:	91 98	1245	STA	(LOWTR),Y	
C70F:	00 F8	1246	BNE	INSRTLIN	
C7E1:	20 7C C9	1247	LINSET	JSR	SETPTRS
C7E4:	A5 67	1248	LDA	TXTTAB	;note que LINSET pode ser chamado por 0(HTM)
C7E6:	A4 68	1249	LDY	TXTTAB+1	
C7E8:	B5 5E	1250	STA	INDEX	
C7EA:	B4 5F	1251	STY	INDEX+1	
C7EC:	18	1252	CLC		
C7ED:	A0 01	1253	ROLINK	LDY	MS01
C7EF:	H1 5E	1254	LDA	(INDEX),Y	
C7F1:	00 08	1255	RNE	PUTLINK	
C7F3:	A5 6F	1256	LDA	WARTAB	
C7F5:	B5 6F	1257	STA	PAGE00	
C7F7:	A5 6A	1258	LDA	WARTAB+1	
C7F9:	B5 60	1259	STA	PAGEEND+1	
C7FB:	4C 28 C7	1260	JMP	NESTART	
C7FE:	A0 04	1261	PUTLINK	LDY	MS04
C800:	C8	1262	FINDEOL	IMY	;Preparar apontadores
C801:	01 5E	1263	LDA	(INDEX),Y	
C803:	00 F8	1264	RNE	FINDEXOL	
C805:	C8	1265	IMY		
C806:	98	1266	TYA		
C807:	65 5E	1267	ADC	INDEX	
C809:	A4	1268	TAX		
C80A:	A0 00	1269	LDY	MS00	
C80C:	91 5E	1270	STA	(INDEX),Y	
C80E:	A5 5F	1271	LDA	INDEX+1	
C810:	69 00	1272	ADC	MS00	
C812:	C8	1273	IMY		
C813:	91 5E	1274	STA	(INDEX),Y	
C815:	06 5E	1275	STX	INDEX	
C817:	05 5F	1276	STA	INDEX+1	
C819:	90 02	1277	BCC	ROLINK	
		1278			
C81B:	A2 00	1279	IMLINK	LDA	MS00
C81D:	06 33	1280	IMLINK2	STX	PROMPT
C81F:	A0 35 04	1281	LDA	TSUTIL	
C822:	00 06	1282	RNE	IMLINK	
C824:	20 6A F0	1283	IMLINK3	JSR	GETLIN
C827:	4C 48 C8	1284	JMP	IMLINKS	;Cont inua no IMLINK
C82A:	A0 F0 9F	1285	IMLINK4	LDA	TSCART
C82D:	C9 01	1286	CHP	MS01	
C82F:	00 F3	1287	RNE	IMLINK3	
C831:	A2 00	1288	LDA	MS00	
C833:	0E 35 04	1289	STX	TSUTIL	
C836:	A9 52	1290	LDA	#"H"	
C838:	90 00 02	1291	STA	IN_X	
C83B:	E8	1292	INX		
C83C:	A9 55	1293	LDA	#"U"	
C83E:	90 00 02	1294	STA	IN_X	
C841:	E8	1295	INX		

CB42:	AF	4E	1296	LD	R"X"	
CB44:	90	00	02	STA	IX,X	
CB47:	EB		1298	INX		
CB48:	ED	EF	1299	INLINS	CPX	00EF ;Terminar linha com SEF
CB4A:	90	02	1300	BCC	00B0FS	;Colocar marc.de fia de linha
CB4C:	A2	EF	1301	LDX	00EF	
CB4E:	A7	00	1302	00B0FS	LD	0000
CB50:	90	00	02	1303	STA	IX,X
CB53:	BA		1304	TXA		
CB54:	F0	00	1305	BEQ	NOI	
CB56:	80	FF	01	1306	STRIP	LD
CB59:	29	7F	1307	AND	007F	;Converter em ASCII
CB5B:	90	FF	01	1308	STA	IX-1,X
CB5E:	CA		1309	DEX		
CB5F:	00	FS	1310	BNE	STRIP	
CB61:	A9	00	1311	NOI	LD	0000
CB63:	A2	FF	1312	LDX	00FF	
CB65:	A6	01	1313	LDY	0001	
CB67:	60		1314	BTS		
			1319	PUT	PARTE10,01	
			131			
CB68:	20	0C	FD	132	INCH	JSR
CB6B:	29	7F		133	AND	007F
CB6D:	60			134	BTS	
				135		
CB6E:	A6	B8		136	GETIN	LDX
CB70:	CA			137	DEX	
CB71:	A0	04		138	LDY	0004
CB73:	84	13		139	STY	DATAFLO
CB75:	24	06		140	RIT	LOCK
CB77:	10	08		141	BPL	PARSE
CB79:	68			142	PLA	
CB7A:	68			143	PLA	
CB7B:	20	7C	C9	144	JSR	SETPTRS
CB7E:	4C	EC	CA	145	JMP	NEWSTY
				146		
CB81:	EB			147	PARSE	INX
CB82:	80	00	02	148	KITCH	LD
CB85:	24	13		149	RIT	DATAFLO
CB87:	70	04		150	BVS	SE
CB89:	C9	20		151	CMP	0"
CB8B:	F0	F4		152	BEN	PARSE
CB8D:	08			153	PHP	
CB8E:	85	0E		154	STA	ENDCH
CB90:	28			155	PLP	
CB91:	C9	22		156	CMP	0""
CB93:	F0	74		157	BEQ	SHIN
CB95:	70	40		158	BVS	PUTIN
CB97:	C9	3F		159	CMP	0"7"
CB99:	00	04		160	BNE	TOK?
CB9B:	A9	8A		161	LD	0print
CB9D:	00	45		162	BNE	PUTIN
CB9F:	C9	30		163	TOK?	CMP
CBAB:	90	04		164	BCC	ISTOK?
CBAD:	C9	3C		165	CMP	0"1"
CBAB:	90	30		166	BCC	PUTIN
CBAB:	84	AD		167	ISTOK?	STY
						STMB62

CBAF: A9 8D	138	LDA	BTCTABL-\$100	
CBAE: 85 9D	139	STA	FAC	
CBAE: A9 C2	140	LDA	BTCTABL-\$100	
CBAF: 85 9E	141	STA	FAC+1	
CBL1: A0 00	142	LDI	#500	
CBB3: 84 0F	143	STY	PNTB	;Guarda token corrente \$80
CBB5: 86	144	DEY		
CBB6: 86 88	145	STX	TXPTRB	
CBB8: CA	146	DEX		
CBBF: CB	147	INY		
CBBA: 00 02	148	BNE	NO	
CBBE: E6 9E	149	IMC	FAC+1	
CBBE: E8	150	IK	INO	
CBBF: 80 00 02	151	LDB	LDA IN,X	
CBC2: C9 20	152	CMF	#'	;Saltar espaços
CBC4: F0 FB	153	BEH	NO	
CBCA: 38	154	SEC		
CBC7: F1 9D	155	SBC	(FAC),Y	;Bate com palav.chave corrente?
CBC9: F0 EE	156	BEH	BT	;Sra.proximo caracter
CBCB: C9 80	157	CMF	#500	;Bate com ult.carac.da palavra-chave
	158	*		
CBCB: 00 41	159	BNE	SKIPTOK	;Mas, passar ao prox.token
CBCF: 05 0F	160	ORA	PNTB	;Obter o token
CBD1: C9 C5	161	CMF	#at	
CBD3: 00 00	162	BNE	PUTTOK	
CBD5: 80 01 02	163	LDA	IN+1,X	
CBD8: C9 4E	164	CMF	#'N'	;Prioridade a ATN
CBD8: F0 3A	165	BEH	SKIPTOK	
CBD8: C9 4F	166	CMF	#'O'	;Prioridade a TO
CBD8: F0 30	167	BEH	SKIPTOK	
CBD8: A9 C5	168	LDA	#at	
CBE2: A4 A0	169	PUTTOK	LDI STRNG2	
CBE4: EB	170	PUTIN	INO	
CBE5: C8	171	INY		
CBE6: 99 FB 01	172	STA	IN-5,Y	
CBE9: 89 FB 01	173	LDA	IN-5,Y	
CBEA: F0 39	174	BEH	DAHE	
CBEA: 38	175	SEC		
CBEF: E9 3A	176	SBC	#'s'	
CBF1: F0 84	177	BEH	SSF	;Desligar DATAFLG
CBF3: C9 49	178	CMF	#ata-'s'	ao fim da instrucao
CBF5: 00 82	179	BNE	SEN?	
CBF7: 85 13	180	SSF	STA DATAFLG	
CBF9: 38	181	SEN?	SEC	
CBF8: E9 78	182	SBC	#ata-'s'	
CBF8: 00 84	183	BNE	KOCHB	
CBF8: 85 0E	184	STA	ENDCHB	;Limpar flag de literais
CY00: 80 00 02	185	SHFTIN	LDA IN,X	
CY03: F0 0F	186	BEH	PUTIN	
CY05: C5 0E	187	CMF	ENDCHB	
CY07: F0 08	188	BEH	PUTIN	
CY09: C8	189	SHEN	INO	
CY0A: 99 FB 01	190	STA	IN-5,Y	
CY0B: EB	191	INO		
CY0E: 00 F0	192	BNE	SHFTIN	;Loop ate term.os literais
CY10: A6 88	193	SKIPTOK	LOX TXPTR	
CY12: EA 0F	194	IMC	PNTB	;Proximo TOKEN

```

C914: 81 90 195 SK2 LDA (FAC),Y ;Saltar palav.chave corrente
C916: C8 196 JNY
C917: 00 02 197 BNE PLU7
C919: E6 9E 198 DMC FAC+1
C91B: 0A 199 PLU7 ASL
C91C: 90 F4 1100 BCC SK2 ;Loop ate saltar a palavr-chav.
C91E: 81 90 1101 LDA (FAC),Y
C920: 00 90 1102 BNE LIM ;Loop ate fim de tabela
1103 * ;palavra-chave
C922: 80 00 02 1104 LDA IN,X ;Nao e palavra chave
C925: 10 88 1105 RPL PUTTOK ;Sempre
C927: 99 F0 D1 1106 DOME STA IN-3,Y ;ERL se modo direto
C92A: C6 89 1107 DEC TXTPTR+1 ;Apostar TXTPTR a IN-1
C92C: A9 FF 1108 LDA #FFF
C92E: 85 8E 1109 STA TXTPTR
C930: 60 1110 RTS
1111
1112 ;Procurar no programa linha cujo num esta agora em LINNUM.
1113 ;Ao sair, carry ligado se encontrada, deslig caso contrar.
1114 ;LOWTR aponta para a linha se foi encontrada, e aponta
1115 ;para a proxima caso contrario
1116
C931: A5 67 1117 FNDLIN LDA TXTTAB ;Comeco da pesquisa
C933: A6 68 1118 LDR TXTTAB+1 no inicio do programa
C935: A0 D1 1119 FLI LDY #D01 ;Comeco da pesquisa em A,X
C937: 85 98 1120 STA LOWTR
C939: 86 9C 1121 STX LOWTR+1
C93B: 81 98 1122 LDA (LOWTR),Y ;Obter apontador
C93D: F0 1F 1123 BEQ MOSUCH ;Desviar se fim de programa
C93F: C8 1124 JNY
C940: C8 1125 JNY
C941: A5 51 1126 LDA LINNUM+1
C943: D1 98 1127 CMP (LOWTR),Y ;Comp.num.da linha (part.alta)
C945: 90 18 1128 BCC RET3 ;Nao, encontrado
C947: F0 03 1129 BEQ FL2
C949: 88 1130 DEY
C94A: 00 09 1131 BNE GETLINK ;Obter proxima linha
C94C: A5 50 1132 FL2 LDA LINNUM
C94E: 88 1133 DEY
C94F: D1 98 1134 CMP (LOWTR),Y ;Numero de linha (baixa ordem)
C951: 90 0C 1135 BCC RET3 ;Passou da linha,nao encontrado
C953: F0 0A 1136 BEQ RET3 ;Encontrado
C955: 88 1137 GETLINK DEY
C956: 81 98 1138 LDA (LOWTR),Y ;Obter proximo apontador
C958: A8 1139 TAX ;(Alta ordem)
C959: 88 1140 DEY
C95A: 81 98 1141 LDA (LOWTR),Y ;Parte baixa ordem
C95C: 80 07 1142 BCS FL1 ;Sempre
C95E: 18 1143 MOSUCH CLC
C95F: 60 1144 RET3 RTS
1145
C960: 00 F0 1146 MEN BNE RET3 ;Desviar se erro de sintaxe
C962: A9 00 1147 SCATCH LDA #500
C964: 85 06 1148 STA LOCK ;Permitir comandos de usuario
C966: A8 1149 TAX
C967: 91 67 1150 STA (TXTTAB),Y
C969: C8 1151 JNY

```

C96A:	91 67	1152	STA	(TXTTAB),Y	
C96C:	A5 67	1153	LDA	TXTTAB	
C96E:	6F 02	1154	ADC	#502	;Se indeterminado,carry
C970:	85 6F	1155	STA	VARTAB	
C972:	85 6F	1156	STA	PRGEND	
C974:	A5 68	1157	LDA	TXTTAB+1	
C976:	6F 00	1158	ADC	#500	
C978:	85 6A	1159	STA	VARTAB+1	
C97A:	85 80	1160	STA	PRGEND+1	
C97C:	20 AE C9	1161	SETPDRS	JSR	STXTPT
C97F:	A9 00	1162	LDA	#500	
C981:	00 2A	1163	CLEAR	BNE	RET4
C983:	A5 73	1164	CLEAR	LDA	NENSI2
C985:	A4 74	1165	LDY	NENSI2+1	
C987:	85 6F	1166	STA	FRETOP	
C989:	84 70	1167	STY	FRETOP+1	
C98B:	A5 69	1168	LDA	VARTAB	
C98D:	A4 6A	1169	LDY	VARTAB+1	
C98F:	85 68	1170	STA	ARTTAB	
C991:	84 6C	1171	STY	ARTTAB+1	
C993:	85 60	1172	STA	STREND	
C995:	84 6E	1173	STY	STREND+1	
C997:	20 69 CB	1174	JSR	RESTORE	
C99A:	A2 55	1175	STRLIN	LOX	#55
C99C:	86 52	1176	STX	TEMPPT	
C99E:	68	1177	PLA		
C99F:	68	1178	TAY		
C9A0:	68	1179	PLA		
C9A1:	A2 F8	1180	LOX	#5F8	;Guardar topo da pilha para
C9A3:	9A	1181	TXS		;apontador e numero da linha
C9A4:	48	1182	PHX		
C9A5:	78	1183	TYA		
C9A6:	48	1184	PHX		
C9A7:	A9 00	1185	LDA	#500	
C9A9:	85 7A	1186	STA	GLDTEXT+1	;CONT normalmente
C9AB:	85 14	1187	STA	SUMFLG	
C9AD:	60	1188	RET4	RTS	
		1189			
C9AE:	18	1190	STXTPT	CLC	
C9AF:	A5 67	1191	LDA	TXTTAB	
C9B1:	6F FF	1192	ADC	#5FF	
C9B3:	85 88	1193	STA	TXTPTR	
C9B5:	A5 68	1194	LDA	TXTTAB+1	
C9B7:	6F FF	1195	ADC	#5FF	
C9B9:	85 89	1196	STA	TXTPTR+1	
C9BB:	60	1197	RETST	RTS	
		1198			
C9BC:	90 04	1199	LIST	BCC	STRTRNG
C9BE:	F0 08	1200	BEQ	STRTRNG	;Ha numero de linha?
C9C0:	C9 C9	1201	CMF	#minus	;Mas
C9C2:	F0 04	1202	BEQ	STRTRNG	;Sim, intervalo de LIST com 0
C9C4:	C9 2C	1203	CMF	#',	
C9C6:	00 C5	1204	BNE	RET4	
C9C8:	20 31 C0	1205	STRTRNG	JSR	LINDET
C9CA:	20 31 C9	1206	JSR	FNOLIN	;LINNUM=começo do intervalo
C9CC:	20 87 00	1207	JSR	CHGOT	;LOWTR aponta a primeira linha
C9CE:	F0 10	1208	BEQ	HALMLST	;Intervalo especificado?
					;Nao, desviar

C903: C9 C9	1209	CMP	MINUS	
C905: FD 04	1210	BEQ	ENDRNG	
C907: C9 2C	1211	CMP	"', "	
C909: 00 04	1212	BNE	NET3	
C90B: 20 B1 00	1213	ENDRNG	JSR	CHRGST ;Atualizar TSTPTR
C90E: 20 31 C0	1214	JSR	LINGET	;LINRUM=fin do intervalo
C9E1: 00 CA	1215	BNE	NET4	;Desviar se erro de sintaxe
C9E3: 68	1216	MADRST	PLA	;Tirar da pilha end.de volta
C9E4: 68	1217		PLA	
C9E5: A5 50	1218	LDA	LINRUM	
C9E7: 05 51	1219	ORA	LINRUM+1	
C9E9: 00 06	1220	BNE	NILST	
C9EB: AF FF	1221	LDA	MSFF	;intervalo maximo
C9ED: 05 50	1222	STA	LINRUM	
C9EF: 05 51	1223	STA	LINRUM+1	
C9F1: A0 01	1224	NILST	LDY	MS01
C9F3: B1 98	1225	LDA	(LOWTR),Y	;Byte alta ordem do apontador
C9F5: FD 44	1226	BEQ	L1STED	
C9F7: 20 7B C0	1227	JSR	ISCHTC	;Testar CONTROL-C
C9FA: 20 20 CE	1228	JSR	CRD0	
C9FD: C8	1229	IMY		
C9FE: B1 98	1230	LDA	(LOWTR),Y	;Obter numero da linha
CA00: AA	1231	TAX		
CA01: C8	1232	IMY		
CA03: B1 98	1233	LDA	(LOWTR),Y	
CA04: C5 51	1234	CMP	LINRUM+1	
CA06: 00 04	1235	BNE	LST0?	
CA08: E4 50	1236	CPX	LINRUM	
CA0A: F0 02	1237	BEQ	LSTLLIN	
CA0C: B0 20	1238	LST0?	BDS	L1STED
CA0E: B4 85	1239	LSTLLIN	STY	FORPNT
CA10: 20 3E E0	1240	JSR	LIMPST	;Imprime X,A
CA13: AF 20	1241	LDA	" "	
CA15: A4 85	1242	L1STLOOP	LDY	FORPNT
CA17: 29 7F	1243	AND	MS7F	
CA19: 20 B1 CE	1244	SENDCR	JSR	OUTD0
CA1C: A5 24	1245	SENDCR	LDA	CH
CA1E: C9 26	1246	CMP	MS26	;Se passou de 26, CR
CA20: 90 07	1247	BCC	NCR	
CA22: 20 20 CE	1248	JSR	CRD0	
CA25: AF 05	1249	LDA	MS05	;E 5 espacos
CA27: B5 24	1250	STA	CH	
CA29: C8	1251	NCR	IMY	
CA2A: B1 98	1252	LDA	(LOWTR),Y	
CA2C: 00 10	1253	BNE	TORLEN?	
CA2E: AA	1254	TAY		;Ao fim da linha,
CA2F: B1 98	1255	LDA	(LOWTR),Y	obter apontador
CA31: AA	1256	TAX		
CA33: C8	1257	IMY		
CA35: B1 98	1258	LDA	(LOWTR),Y	
CA37: B6 98	1259	STX	LOWTR	;Apontar a proxima linha
CA39: B5 9C	1260	STA	LOWTR+1	
CA3B: 00 B4	1261	BNE	NILST	
CA3D: AF 00	1262	L1STED	LDA	MS00
CA3F: 20 B1 CE	1263	JSR	OUTD0	;CR e fim
CA40: AC EC CA	1264	JMP	MEMSTT	
	1265			

CA43:	08	1266	GETCHR	BNY		;Obter caracter da tabela
CA44:	00 02	1267		BNE	SC	
CA46:	E6 9E	1268		INC	FAC+1	
CA48:	B1 90	1269	GC	LDA	(FAC),Y	
CA4A:	60	1270		RTS		
		320		PUT	PARTESE,01	
CA4B:	10 CC	12	TOKEN?	BPL	SEMCHRG	;Desviar se nao for token
CA4D:	4C 3B F7	12		JMP	BUSTOKEN	
CA50:	38	13	CONTIN	SEC		
CA51:	E9 7F	14		SBC	#57F	;Fazer pointer a tabela
CA53:	AA	15		TAX		
CA54:	B4 85	16		STY	FORPNT	;Guardar pointer a linha
CA56:	A0 80	17		LDY	#TOKTABL-\$100	
CA58:	B4 90	18		STY	FAC	;Fazer FAC apontar a tabela
CA5A:	A0 C2	19		LDY	#TOKTABL-\$100	
CA5C:	B4 9E	110		STY	FAC+1	
CA5E:	A0 FF	111		LDY	#\$FF	
CA60:	CA	112	SKPTX	DEX		;Contar tokens ate X
CA61:	F0 07	113		BEQ	PRTOX	
CA63:	20 43 C6	114	TOKLP	JSR	GETCHR	
CA64:	10 FB	115		BPL	TOKLP	
CA66:	30 F6	116		BNE	SKPTX	
CA6A:	A9 20	117	PRTOX	LDA	#' '	;Token encontrado,mandar espaco
CA6C:	20 B1 CE	118		JSR	OUTDO	
CA6F:	20 43 C6	119	TOKLUP	JSR	GETCHR	;E token
CA72:	30 05	120		RAC	TOKDONE	
CA74:	20 B1 CE	121		JSR	OUTDO	
CA77:	00 F6	122		BNE	TOKLUP	
CA79:	20 B1 CE	123	TOKDONE	JSR	OUTDO	;Enviar ultimo carac.do token
CA7C:	A9 20	124		LDA	#' '	;Mandar espaco final
CA7E:	00 95	125		BNE	LISTLOOP	;Voltar a linha
		126				
		127	*FOR coloca os seguintes 18 bytes na pilha:			
		128	a TXTPTR			
		129	a Numero da linha			
		130	a Valor da variavel FOR (ponto flutuante 5 bytes)			
		131	a Sinal do STEP			
		132	a Valor do STEP (5 bytes)			
		133	a FORPNT (pointer a variavel)			
		134	a "Token" de FOR			
		135				
CA80:	A9 80	136	FOR	LDA	#\$00	
CA82:	05 14	137		STA	SUBFL6	;Indices nao sao permitidos
CA84:	20 68 CD	138		JSR	LET	
CA87:	20 54 C6	139		JSR	BTFRPNT	;Variavel FOR esta ativa?
CA8A:	00 05	140		BNE	FOR2	;Desviar se nao
CA8C:	BA	141		TXA		;Sim,cancela-e as seguintes
CA8D:	A9 0F	142		ADC	#\$0F	
CA8F:	AA	143		TAX		
CA90:	9A	144		TXS		
CA91:	AA	145	FOR2	PLA		
CA92:	68	146		PLA		
CA93:	A9 09	147		LDA	#\$09	
CA95:	20 C5 C6	148		JSR	CHRGX	;Verf.se stack pointer != \$48
CA98:	20 C8 CC	149		JSR	DATAN	;Apontar ao comando seguinte
CA9B:	08	150		CLE		;Colocar este end.no stack
CA9C:	98	151		TYA		

CAP0: 63 88	>52	ADC	TXTPTR	
CAP1: 48	>53	PHA		
CAP0: A5 89	>54	LDA	TXTPTR+1	
CAP2: 69 90	>55	ADC	#500	
CAP4: 48	>56	PHA		
CAP5: A5 76	>57	LDA	CURLIN+1	
CAP7: 48	>58	PHA		
CAP8: A5 75	>59	LDA	CURLIN	
CAP4: 48	>60	PHA		
CAP8: A9 C1	>61	LDA	#to	
CAP0: 20 E8 D1	>62	JSR	STANCH	
CAP0: 20 92 00	>63	JSR	CHIRAM	
CAP3: 23 8F D8	>64	JSR	FINNUM	
CAP6: A5 A2	>65	LDA	FACSEM	
CAP8: 09 7F	>66	BRA	#577	
CAP4: 25 9E	>67	AND	FAC+1	
CAPC: 85 9E	>68	STA	FAC+1	
CAP6: A9 C9	>69	LDA	BSTEP	;Preparar para retorno a STEP
CAC0: A0 CA	>70	LDY	RISTEP	
CAC2: 85 5E	>71	STA	INDEX	
CAC4: 84 5F	>72	STY	INDEX+1	
CAC6: 4C 4B D1	>73	JMP	FLUSHAC	;Retorno a STEP
CAC9: A9 20	>74	LDA	#HUM	;STEP default = 1
CAC8: A0 DC	>75	LDY	#HUM	
CAC0: 20 13 DE	>76	JSR	NOVFN	
CAC0: 20 87 00	>77	JSR	CHRGOT	
CAC0: C9 C7	>78	CMF	#step	
CAC5: 00 D6	>79	BNE	ONESTEP	
CAC7: 20 81 D0	>80	JSR	CHRGOT	;Foi especificado um step,
CAC4: 20 8F D0	>81	JSR	FINNUM	;Note-lo
CAC0: 20 9C 8C	>82	JSR	SIGN	
CAC0: 20 30 D1	>83	JSR	PSHFACE	
CAC3: A5 B6	>84	LDA	FOFPNT+1	
CAC5: 48	>85	PHA		
CAC6: A5 B5	>86	LDA	FOFPNT	
CAC8: 48	>87	PHA		
CAC9: A9 81	>88	LDA	#for	
CAC8: 48	>89	PHA		
CAC6: B6	>90	NEGSTT	TSX	;Executar novo comando
CAC0: 86 F8	>91	STX	RENSTX	
CACF: 20 78 C8	>92	JSR	ISCONT	
CAC2: A5 B8	>93	LDA	TXTPTR	
CAC4: A4 89	>94	LDY	TXTPTR+1	
CAC6: A6 76	>95	LDX	CURLIN+1	;Modo direto
CAC8: EB	>96	BHX		
CAC9: F0 04	>97	BER	DIRECT	;Desviar se sim
CAC8: 85 79	>98	STA	OLDTEXT	;Guardar TXTPTR se estiver no
CAC0: 84 7A	>99	STY	OLDTEXT+1	programa para CONT eventual
CACF: A0 D0	>100	LDY	#500	
CRC0: 81 88	>101	LDA	(TXTPTR),Y	;Fim da linha?
CRC3: 00 50	>102	BNE	COLON?	;Nao, '?' ?
CRC5: A0 02	>103	LDY	#502	;Sim, link 0?
CRC7: 81 88	>104	LDA	(TXTPTR),Y	
CRC9: 18	>105	CLC		
CRC4: F0 34	>106	REB	GOEND	;Fim se linha 0
CRC0: C8	>107	DNY		
CRC0: 81 B8	>108	LDA	(TXTPTR),Y	

CB0F: 85 75	3109	STA	CURLIN	;Mao, e fim,
CB11: 00	3110	DAY		
CB12: 81 88	3111	LDA	(TXPTR),Y	;guardar o numero da linha
CB14: 85 76	3112	STA	CURLIN+1	
CB16: 98	3113	TXA		
CB17: 43 88	3114	ADC	TXPTR	;Posic. pointer ao texto TXPTR
CB19: 85 88	3115	STA	TXPTR	
CB1B: 90 02	3116	BCC	TRACE?	
CB1D: E6 89	3117	INC	TXPTR+1	
CB1F: 24 F2	3118	TRACE?	BIT	TROFLE
CB21: 10 14	3119	RPL	EXECUTE	;Foi pedido trace?
CB23: A6 76	3120	LOX	CURLIN+1	;Mao,desviar
CB25: E8	3121	INX		
CB26: F0 0F	3122	BEQ	EXECUTE	;Saltar se comando direto
CB28: 89 23	3123	LDA	"B"	;Imprime "B"
CB2A: 20 81 CE	3124	JSR	OUTDO	
CB2D: A6 75	3125	LOX	CURLIN	
CB2F: A5 76	3126	LDA	CURLIN+1	
CB31: 20 3E E0	3127	JSR	LINPST	;E o numero
CB34: 20 7C CE	3128	JSR	OUTSP	
CB37: 20 81 00	3129	EXECUTE	JSR	CHRGET
CB3A: 20 42 CB	3130	JSR	GOCHO	;Obter primeiro caract.do comando
CB3D: 4C CC CA	3131	JMP	NEWSTT	;Comeca a processar a linha
	3132			;Retornar a outro comando
CB40: F0 60	3133	GOCHO	BEQ	END4
CB42: F0 33	3134	GOCHO	BEQ	SETS
CB44: E9 80	3135	GOCHO2	SBC	#580
CB46: 90 17	3136	ROC	MOTOK	;Token?
CB49: C9 40	3137	CMP	#540	;Desviar se nao for
CB4B: 90 06	3138	BCC	GOCHO3	;Token tipo rotina?
CB4C: C9 68	3139	CMP	#568	;Erro de sintaxe se nao for
CB4E: 90 16	3140	BCC	JST	
CB50: E9 12	3141	SBC	#512	
CB52: 0A	3142	GOCHO3	ASL	;Obter endereco da rotina
CB53: A8	3143	TAY		
CB54: 89 07 C2	3144	LDA	CHTABL+1,Y	
CB57: 48	3145	PHA		;E coloca-la no stack
CB59: 89 06 C2	3146	LDA	CHTABL,Y	
CB5B: 48	3147	PHA		
CB5C: 4C 81 00	3148	JMP	CHRGET	;Obter proximo caract.
CB5F: 4C 68 CD	3149	JMP	LET	;Atribuiçao a variav.(A48)
CB62: C9 3A	3150	COLW?	CMP	"I"
CB64: F0 89	3151	BEQ	TRACE?	
CB66: 4C F1 D1	3152	JST	JMP	SYNERR
	3153			
CB6F: 38	3154	RESTORE	SEC	
CB6A: A5 67	3155	LDA	TXTHW	
CB6C: E9 D1	3156	SBC	#501	
CB6E: A4 68	3157	LDT	TXTHW+1	
CB70: B0 D1	3158	BCC	SETDA	
CB72: 88	3159	DEY		
CB73: 85 70	3160	SETDA	STA	OUTPTR
CB75: 84 7E	3161	STY	OUTPTR+1	
CB77: 60	3162	RETS	RTS	
	3163			
CB79: 84 EF	3164	ISCHTC	STY	YCHTC
CB7A: 20 42 F0	3165	JSR	SCALL	

CB7D:	44 EF	1166	LDI	YONTC	
CB7F:	EA	1167	WOP		
CB80:	C9 83	1168	CMR	MS83	
CB82:	F0 D1	1169	BEQ	OK	
CB84:	60	1170	RTS		
CB85:	20 68 CB	1171	BK	JSR INCHR	
CB8B:	A2 FF	1172	ERRFLG?	LDX	MSFF ;Foi testado CONTROL-C
CB8A:	24 08	1173	BIT	ERRFLG	
CB8C:	10 03	1174	BPL	CTRC?	
CB8E:	4C 4A E6	1175	JMP	HANDLER	
CB91:	C9 03	1176	CTRC?	CMR	MS03
		1177			
CB93:	80 D1	1178	STOP	BOS	END2
		1179			
CB95:	18	1180	END	CLC	
CB96:	00 3C	1181	END2	BNE	RET6
CB98:	A5 88	1182	LDA	TXTPTR	
CB9A:	A4 89	1183	LDY	TXTPTR+1	
CB9C:	A6 76	1184	LDX	CURLIN+1	;Foi direto?
CB9E:	EB	1185	INX		
CB9F:	F0 0C	1186	BEQ	END3	;Sim,desviar
CBAl:	85 79	1187	STA	OLDTEXT	
CBAB:	84 78	1188	STY	OLDTEXT+1	
CBAC:	A5 75	1189	LDA	CURLIN	
CBAD:	84 76	1190	LDY	CURLIN+1	
CBAF:	85 77	1191	STA	OLDLIN	
CBAB:	84 78	1192	STY	OLDLIN+1	
CBAD:	60	1193	END3	PLA	
CBAE:	68	1194	PLA		
CBAF:	A9 4C	1195	END4	LDA	BBREAKIN
CBBl:	A0 C6	1196	LDY	BBREAKIN	
CB83:	90 03	1197	BCC	GOSTART	
CB85:	4C 20 C7	1198	JMP	PRINTM?	
CB88:	4C 28 C7	1199	JMP	RESTART	
		1200	PUT	PARTSELF_D1	
		1201			
CB8B:	D0 17	1202	CONT	BNE	RET6
CB8D:	A2 03	1203	LDX	BDMOSSBL-MEMSEIRD	
CB8F:	A4 7A	1204	LDY	OLDTEXT+1	
CB81:	D0 03	1205	BNE	CON	
CB83:	4C 01 C7	1206	JMP	ERROR	
CB86:	A5 79	1207	LDA	OLDTEXT	
CB88:	85 88	1208	STA	TXTPTR	
CB8A:	84 89	1209	STY	TXTPTR+1	
CB8C:	A5 77	1210	LDA	OLDLIN	
CB8E:	44 78	1211	LDY	OLDLIN+1	
CB80:	85 75	1212	STA	CURLIN	
CB82:	84 76	1213	STY	CURLIN+1	
CB84:	60	1214	RET6	RTS	
		1215			
CB85:	38	1216	SAVE	SEC	
CB86:	A5 AF	1217	LDA	PROEND	
CB88:	E3 67	1218	SBC	TXTTAN	
CB8A:	85 50	1219	STA	LIDRUM	
CB8C:	A5 80	1220	LDA	PROEND+1	
CB8E:	E3 68	1221	SBC	TXTTAN+1	
CB80:	85 51	1222	STA	LIDRUM+1	

0RE2: 20 15 CC 120	JSR	WRTIO	
0RE3: 20 00 FE 124	JSR	WRITE	
0RE8: 20 26 CC 125	JSR	PROG10	
0RE9: 4C 00 FE 126	JMP	WRITE	
0RE: 20 15 CC 127	LOW	JSR	WRTIO
0RF1: 20 00 FE 128	JSR	READM	
0RF4: 18 129	CLC		
0RF5: A5 67 130	LDA	TEXTAB	
0RF7: A5 50 131	ADC	LINNUM	
0RF9: 85 69 132	STA	WRTAB	
0RF8: A5 68 133	LDA	TEXTAB+1	
0RFD: A5 51 134	ADC	LINNUM+1	
0RFF: 85 6A 135	STA	WRTAB+1	
0C01: A5 52 136	LDA	TEMPPT	
0C03: 85 04 137	STA	LOCK	
0C05: 20 26 CC 138	JSR	PROG10	
0C08: 20 00 FE 139	JSR	READM	
0C08: 24 04 140	BIT	LOCK	;Se negativo ler cassete
0C08: 10 03 141	RPL	JUNK	
0C0F: 4C 7C C9 142	JMP	SETPTRS	
0C12: 4C E1 C7 143	JUNK	JMP	LINWSET
			144
0C15: A9 50 145	WRTIO	LDA	BLINNUM
0C17: A0 00 146	LDY	#000	
0C19: 85 3C 147	STA	ASL	
0C1A: 84 39 148	STY	ALH	
0C1B: A9 52 149	LDA	RTMPPT	
0C1F: 85 3E 150	STA	A2L	
0C21: 84 3F 151	STY	A2H	
0C23: 84 04 152	STY	LOCK	
0C25: 60 153	RTS		
			154
0C26: A5 67 155	PROG10	LDA	TEXTAB
0C28: A4 68 156	LDY	TEXTAB+1	
0C2A: 85 3C 157	STA	AIL	
0C2C: 84 30 158	STY	ALH	
0C2E: A5 69 159	LDA	WRTAB	
0C30: A4 6A 160	LDY	WRTAB+1	
0C32: 85 3E 161	STA	A2L	
0C34: 84 3F 162	STY	A2H	
0C36: 60 163	RTS		
			164
0C37: 00 165	RUN	PHP	
0C38: C6 76 166	DEC	CURLIN+1	
0C3A: 20 167	PLP		
0C3B: 00 03 168	BNE	ENDLINE	;Desviar se ha numero de linha
0C3C: 4C 7C C9 169	JMP	SETPTRS	;Especificar comeco do programa
0C40: 20 03 C9 170	ENDLINE	JSR	CLEARC
0C43: 4C 5A CC 171		JMP	GOLINE
			;Desviar a linha especificada
			172
			173
			174
			175
			176
			177
			178
0C46: A9 00 179	GOSUB	LDA	#000

*GOSUB deixa na pilha:
 * Endereco de volta (ROMSTT)
 * TEXTPT
 * Numero da linha
 * "Token" do GOSUB

CC4B: 20 C5 C6 180	JSR	CHREN	;Pointer da pilha e !=536?
CC4B: A5 B7 181	LDA	TXTPTR+1	
CC4D: 4B 182	PMA		
CC4E: A5 B8 183	LDA	TXTPTR	
CC50: 4B 184	PMA		
CC51: A5 76 185	LDA	CURLIN+1	
CC53: 4B 186	PMA		
CC54: A5 75 187	LDA	CURLIN	
CC56: 4B 188	PMA		
CC57: A9 B0 189	LDA	Rgesch	
CC59: 4B 190	PMA		
CC5A: 20 B7 B0 191	GOLINE	JSR	CHRGOT
CC5B: 20 A3 CC 192	JSR	GOTO	
CC60: 4C EC DA 193	JMP	MEMSTT	
			194
CC63: 20 31 C3 195	GOTO	JSR	LINRET
			;Obter linha do GOTO
CC66: 20 C8 CC 196	JSR	REMI	;T Apontando ao fim da linha
CC6F: A5 76 197	LDA	CURLIN+1	;Pag.corrente (pagina do GOTO
CC6B: C5 51 198	CMF	LINNUM+1	
CC6D: B0 B0 199	BCC	G01	;Nao, pesq. do comeco do progr.
CC6F: 9B 1100	T1A		;Sim, pesq. da proxima linha
CC70: 3B 1101	SEC		
CC71: A5 B8 1102	ADC	TXTPTR	
CC73: A6 B9 1103	LOX	TXTPTR+1	
CC75: 9B 07 1104	BCC	G02	
CC77: B8 1105	INX		
CC7B: B0 B4 1106	BCC	G02	
CC7A: A5 67 1107	LDA	TXTAB	;Obter comeco do programa
CC7C: A6 A8 1108	LOX	TXTAB+1	
CC7E: 20 35 C9 1109	G02	JSR	FL1
			;Procurar linha do GOTO
CC81: 90 1E 1110	BCC	UNDER	;Erro: linha nao encontrada
CC83: A5 9B 1111	LDA	LOWTR	;TXTPTR apontando a linha GOTO
CC85: E9 01 1112	SBC	#501	
CC87: B5 B8 1113	STA	TXTPTR	
CC89: A5 9C 1114	LDA	LOWTR+1	
CC8B: E9 00 1115	SBC	#500	
CC8D: B5 B9 1116	STA	TXTPTR+1	
CC8F: 60 1117	RET7	RTS	;Retornar a MEMSTT ou G02UN
			1118
CC90: B0 F0 1119	POP	BNE	RET7
CC92: A9 FF 1120	LDA	#FFF	
CC94: B5 B5 1121	STA	FORPNT	
CC96: 20 54 C6 1122	JSR	STFORPNT	
CC9F: 9A 1123	TXS		
CC9A: C9 B3 1124	CMF	Rgesb	;E ultimo GOSUB?
CC9C: F0 B8 1125	BEQ	RETURN	
CC9E: A2 14 1126	LOX	RTNSeGSR-MENSERR0	
CCA0: 2C 1127	HEX	2C	
CCA1: A2 55 1128	UNDER	LOX	RCORWDEF-MENSERR0
CCA3: 4C D1 C7 1129	JMP	ERR0R	
			1130
CCA6: 4C F1 D1 1131	SEYMER	JMP	SYMER
			1132
CCAF: 6B 1133	RETURN	PLA	
CCAH: 6B 1134		PLA	
CCAB: C0 42 1135	CPY	Rpy#2	
CCAD: F0 3B 1136	BEQ	PULL3	;Desviar se POP

CCAF: 05 75	1137	STA	CURLIN	Obter numero da linha
CCB1: 68	1138	PLA		
CCB2: 05 74	1139	STA	CURLIN+1	
CCB4: 68	1140	PLA		
CCB5: 05 00	1141	STA	TXTPTR	
CCB7: 68	1142	PLA		
CCB8: 05 0F	1143	STA	TXTPTR+1	
CCBA: 20 C8 CC	1144	DATA	JSR DATAM	Passar ao proximo comando
CCB0: 90	1145	ADDOH	TYA	
CCBE: 18	1146	CLC		
CCBF: 45 00	1147	ADC	TXTPTR	
CCC1: 05 00	1148	STA	TXTPTR	
CCC3: 90 02	1149	RCC	RETB	
CCC5: E6 0F	1150	JNC	TXTPTR+1	
CCC7: 40	1151	RETB	RTS	
	1152			
CCC9: A2 3A	1153	DATAM	LDX B*1	Fazer Y = deslocamento
CCCA: 2C	1154	HEX	2C	ate fim de linha ou "1"
CCCB: 42 00	1155	BEH	LDX #000	
CCCD: 06 00	1156	STX	CHWAC	
CCCF: 40 00	1157	LDY	#000	
CCD1: 04 0E	1158	STY	ENDCHR	
CCD3: A5 0E	1159	RH1	LDA ENDCHR	Para contar pares de apostrofe
CCD5: A6 00	1160	LDX	CHWAC	
CCD7: 05 00	1161	STA	CHWAC	
CCD9: 06 0E	1162	STX	ENDCHR	
CCDB: 01 00	1163	RH2	LDA (TXTPTR),Y	
CCDD: F0 0A	1164	BEB	RETB	Se fim de linha,
CCDF: C5 0E	1165	CMF	ENDCHR	sair com Y=deslocamento
CCE1: F0 04	1166	BEB	RETB	
CCE3: C0	1167	DNY		
CCE4: C9 22	1168	CMF	B*1	
CCE6: 00 F3	1169	RNE	RH2	Se nao for apostrofe,continuar
CCE8: F0 0F	1170	BEH	RH1	Trocar par e continuar
	1171			
CCEA: 48	1172	PULL3	PLA	
CCEB: 48	1173		PLA	
CCEC: 48	1174		PLA	
CCE0: 40	1175		RTS	
	1176			
CCEE: 20 A0 00	1177	IF	JSR FINEAL	
CCF1: 20 07 00	1178	JSR	CHGOT	
CCF4: C9 A8	1179	CMF	Agoto	
CCF6: F0 05	1180	BEH	TRUE?	
CCF8: AF C4	1181	LDA	Bthen	
CCFA: 20 E0 01	1182	JSR	SYNCHR	
CCFD: A5 90	1183	TRUE?	LDA FAC	Condicao verdadeira ou falsa?
CCFF: 00 05	1184	RNE	IFTRUE	Se verdadeira,desviar
CD01: 20 C8 CC	1185	BEH	JSR BEH	Saltar resto da linha
CD04: F0 07	1186	BEH	ADDOH	Desviar sempre
	1187			
CD06: 20 07 00	1188	IFTRUE	JSR CHGOT	Comando ou numero?
CD09: 00 03	1189	BOS	JGOCNO	Caso comando
CD0B: 4C 43 CC	1190	JMP	GOTO	Caso numero
CD0E: 4C 42 C8	1191	JGOCNO	JMP GOCNO	Executar o comando
	1192			
CD11: 20 12 0A	1193	ONGOTO	JSR GETRYT	Obter numero em FAC+4

CD14:	48	1194	PHA	
CD15:	C9 80	1195	CMF	Bgssub
CD17:	F0 04	1196	BEQ	DNCKT
CD19:	C9 A8	1197	CMF	Bgate
CD1B:	00 89	1198	BNE	SSYMER
CD1D:	C6 A1	1199	DNCKT	;Chegou?
CD1F:	00 04	1200	BNE	NUMM
CD21:	68	1201	PLA	;Nao,continuar pesquisando
CD22:	4C 44 C8	1202	JMP	GOCHO2
CD25:	20 81 00	1203	NUMM	;Sim, obter comando e executar
CD26:	20 35 C0	1204	JSR	CHRGET
CD28:	C9 2C	1205	CMF	B', "
CD2B:	F0 EE	1206	BEQ	DNCKT
CD2F:	68	1207	PLA	;Bao encontrado,ignorar comando
CD30:	40	1208	RET9	RTS
		322	PUT	PARTE16,01
		31		
CD31:	A2 00	12	LINGET	LDX #500
CD33:	86 50	13	STX	LINMM
CD35:	86 51	14	STX	LINMM+1
CD37:	80 F7	15	ASCHEX	BCS RET9
CD39:	E9 2F	16	SBC	#10'-1
CD3B:	85 00	17	STA	CHARAC
CD3D:	A5 51	18	LDA	LINMM+1
CD3F:	85 5E	19	STA	INDEX
CD41:	C9 19	110	CMF	BSF#10
CD43:	80 04	111	BCS	DOT0?
CD45:	A5 50	112	LDA	LINMM
CD47:	0A	113	ASL	
CD4B:	26 5E	114	ROL	INDEX
CD4A:	0A	115	ASL	
CD4B:	26 5E	116	ROL	INDEX
CD4D:	A5 50	117	ADC	LINMM
CD4F:	85 50	118	STA	LINMM
CD51:	A5 5E	119	LDA	INDEX
CD53:	A5 51	120	ADC	LINMM+1
CD55:	85 51	121	STA	LINMM+1
CD57:	06 50	122	ASL	LINMM
CD59:	26 51	123	ROL	LINMM+1
CD5B:	A5 50	124	LDA	LINMM
CD5D:	A5 00	125	ADC	CHARAC
CD5F:	85 50	126	STA	LINMM
CD61:	90 02	127	BCZ	MOD18
CD63:	E6 51	128	INC	LINMM+1
CD65:	20 81 00	129	MOD18	JSR CHRGET
CD6B:	4C 37 C0	130	JMP	ASCHEX
		131		
CD6B:	20 03 00	132	LET	JSR PTRGET
CD6E:	85 85	133	STA	FORPNT
CD70:	84 84	134	STY	FORPNT+1
CD72:	A9 00	135	LDA	#equal
CD74:	20 E8 04	136	JSR	SYNCR
CD77:	A5 12	137	LDA	VALTYP+1
CD79:	48	138	PHA	
CD7A:	A5 11	139	LDA	VALTYP
CD7C:	48	140	PHA	
CD7D:	20 A3 00	141	JSR	FIMEXL

CD00:	68	142	PLA	
CD01:	26	143	ROL	;Carry=sinal VOLT?
CD02:	20 95 00	144	JSR	CHVAL
CD05:	00 18	145	BNE	LETSTR ;Se string de caracteres
CD07:	68	146	PLA	
CD08:	10 12	147	LET2	BPL LETREAL
CD0A:	20 0C 0E	148	JSR	RND0 ;Variavel inteira
CD0C:	20 26 04	149	JSR	ATINT
CD10:	A0 00	150	LDY	#000
CD12:	A5 A0	151	LDA	VPNT
CD14:	91 05	152	STA	(FORPNT),Y
CD16:	C8	153	INT	
CD17:	A5 A1	154	LDA	VPNT+1
CD19:	91 05	155	STA	(FORPNT),Y
CD1B:	60	156	RTS	
CD1C:	4C 41 0E	157	LETREAL	JMP SETFOR
CD1F:	68	158	LETSTR	PLA
CD20:	A0 02	159	PUSSTR	LDY #02
CD22:	B1 A0	160	LDA	(VPNT),Y
CD24:	C5 70	161	CMF	FRETOP+1
CD26:	90 17	162	BCC	COPSTR ;Nao esta na area de strings
CD28:	00 07	163	BNE	DESC?
CD2A:	88	164	DEY	
CD2C:	B1 A0	165	LDA	(VPNT),Y
CD2E:	C5 4F	166	CMF	FRETOP
CD2F:	90 0E	167	BCC	COPSTR
CD31:	A4 A1	168	DESC?	LDY VPNT+1 ;Tem descritor?
CD33:	C4 6A	169	CMF	VPNTAB+1
CD35:	90 06	170	BCC	COPSTR ;Sim,copiar
CD37:	00 00	171	BNE	MEMDESC
CD39:	A5 A0	172	LDA	VPNT
CD3B:	C5 69	173	CMF	VPNTAB
CD3D:	60 07	174	BCS	MEMDESC
CD3F:	A5 A0	175	COPSTR	LDY VPNT ;Copiar somente o descritor
CD41:	A4 A1	176	LDY	VPNT+1
CD43:	4C 0C 0D	177	JMP	COPY
CD44:	A0 00	178	MEMDESC	LDY #00 ;Criar novo descritor
CD48:	B1 A0	179	LDA	(VPNT),Y
CD4A:	20 EF 06	180	JSR	STR IN
CD4C:	A5 0C	181	LDA	DSOPT
CD4F:	A4 00	182	LDY	DSOPT+1
CD51:	B5 A8	183	STA	STRING1
CD53:	B4 AC	184	STY	STRING1+1
CD55:	20 EE 06	185	JSR	MOVING
CD5B:	A9 90	186	LDA	BFAC
CD5D:	A0 00	187	LDY	#000
CD5F:	B5 0C	188	COPY	STA DSOPT
CD61:	B4 00	189	STY	DSOPT+1
CD63:	20 4F 09	190	JSR	FRETHG
CD65:	A0 00	191	LDY	#000
CD67:	B1 0C	192	LDA	(DSOPT),Y
CD69:	91 05	193	STA	(FORPNT),Y
CD6B:	C8	194	INT	
CD6D:	B1 0C	195	LDA	(DSOPT),Y
CD6F:	91 05	196	STA	(FORPNT),Y
CD71:	C8	197	INT	
CD73:	B1 0C	198	LDA	(DSOPT),Y

CE1: 91 85	199	STA	(FORPNT),Y	
CE3: 60	1100	RTS		
	1101			
CE4: 20 62 CE	1102	PRSTRNG	JSR	STRPAT
CE7: 20 87 00	1103	JSR	CHRGOT	
CE6: F0 24	1104	PRINT	BEQ	CRDO ;Desviar se fim do comando
CEFC: F0 29	1105	PRINT2	BEQ	NET10
CEFE: C9 C0	1106	CMF	Rtab	
CE00: F0 39	1107	BEQ	TABWHERE	
CE02: C9 C3	1108	CMF	"Aspc	
CE04: 18	1109	CLC		
CE05: F0 34	1110	BEQ	TABWHERE	
CE07: C9 2C	1111	CMF	"','	
CE09: 18	1112	CLC		
CE0A: F0 1C	1113	BEQ	TAB	
CE0C: C9 38	1114	CMF	"';'	
CE0E: F0 44	1115	BEQ	NEXTCHR	
CE10: 20 A3 00	1116	JSR	FRMVL	;Analisar formula
CE13: 24 11	1117	BIT	VALTYP	
CE15: 30 00	1118	RRI	PRSTRNG	;Desviar se string
CE17: 20 4E ED	1119	JSR	FOUT	;Conv.numero em FAC a string
CE1A: 20 01 07	1120	JSR	STNLT	;Criar descritor temporario
CE1D: 4C F4 C0	1121	JMP	PRSTRNG	;Imprimir
	1122			
CE20: A9 00	1123	CRDO	LDA	WS00
CE22: 20 81 CE	1124	JSR	OUTBO	
CE25: A9 FF	1125	NEGATE	EOR	MSFF
CE27: 60	1126	NET10	RTS	
	1127			
CE2B: A5 24	1128	TAB	LDA	CH
CE2A: C9 18	1129	CMF	MS10	
CE2C: 90 05	1130	BCC	NXELN	
CE2E: 20 20 CE	1131	JSR	CRDO	
CE31: 00 21	1132	BNE	NEXTCHR	;Sempre
CE33: A9 10	1133	NXELN	ADC	MS10
CE35: 29 F0	1134	AND	MSFD	;Tab 16,32
CE37: 85 24	1135	STA	CH	
CE39: 90 19	1136	BCC	NEXTCHR	;Sempre
	1137			
CE3B: 08	1138	TABWHERE	PHP	;Restaurar casos SPC ou TAB
CE3C: 20 0F DA	1139	JSR	STRYTC	
CE3F: C9 29	1140	CMF	"')"	
CE41: F0 03	1141	BEQ	SPC?	
CE43: 4C F1 01	1142	JMP	SYNMR	
CE46: 28	1143	PLP		
CE47: 90 07	1144	BCC	TABIT	;Desviar se SPC
CE49: CA	1145	DEX		
CE4A: 8A	1146	TGA		
CE4B: E5 24	1147	SRC	CH	;Calc.num.de espacos a andar
CE4D: 90 05	1148	BCC	NEXTCHR	;Caso negativo
CE4F: AA	1149	TAX		
CE50: 08	1150	TABIT	INX	
CE51: CA	1151	NXSPC	DEX	
CE52: 00 06	1152	BNE	OSPC	
CE54: 20 81 00	1153	NEXTCHR	JSR	CHRGOT ;Testar fim de comando
CE57: 4C FC C0	1154	JMP	PRINT2	
CE5A: 20 7C CE	1155	OSPC	JSR	OUTSP

CE30:	00 F2	1156	BNE	NO\$PC	;Sempre
		1157			
CE3F:	20 01 07	1158	STROUT	JSR	STRLIT ;Imprimir string em (A,Y)
CE42:	20 1A 09	1159	STRPRT	JSR	FREFAC ;Obter pointer ao string
CE45:	AA	1160	TAX		;Comprimento
CE4A:	A0 00	1161	LDY	#500	
CE4B:	ED	1162	DAX		
CE4F:	CA	1163	WCHNR	DEX	
CE5A:	FB 88	1164	BEQ	RET10 ;Sair se fim de string	
CE5C:	B1 5E	1165	LDA	(INDEX),Y	
CE5E:	20 B1 CE	1166	JSR	OUTDO	
CE71:	C8	1167	INY		
CE72:	C9 00	1168	CHP	#500	
CE74:	D0 F3	1169	BNE	NO\$CHNR	
CE76:	20 25 CE	1170	JSR	NEGATE	
CE79:	4C 69 CE	1171	JMP	WCHNR	
		1172			
CE7C:	A9 20	1173	OUTSP	LDA	#' '
CE7E:	2C	1174	HEX	2C	
CE7F:	A9 3F	1175	OUTQUES	LDA	#'''
CE81:	09 80	1176	OUTDO	ORA	#80
CE83:	C9 A0	1177	CHP	#"	;Caracter de controle?
CE85:	90 02	1178	BCC	SEMO	;Sim,desviar
CE87:	05 F3	1179	ORA	ORMASK	;Converter ou nao mudar
CE8F:	20 ED F0	1180	SEMO	JSR	CDUT
CE9C:	29 7F	1181	AND	#57F	
CEBE:	48	1182	PHA		
CEBF:	A5 F1	1183	LDA	SPEEDZ	
CE91:	20 A8 FC	1184	JSR	WAIT	
CE94:	68	1185	PLA		
CE95:	60	1186	RTS		
		1187			
CE9A:	A5 15	1188	INPUTERR	LDA	INPUTFLG
CE9B:	F0 12	1189	BEQ	RESPERR	;Se INPUT
CE9A:	30 04	1190	BNE	READERR	;Se READ
CE9C:	A0 FF	1191	LDY	#FF	;Se GET
CE9E:	D0 04	1192	BNE	ERLIN	
CEA0:	A5 78	1193	READERR	LDA	DATLIN
CEA2:	A4 7C	1194	LDY	DATLIN+1	
CEA4:	B5 75	1195	ERLIN	STA	CURLIN
CEA6:	B4 76	1196	STY	CURLIN+1	
CEAB:	4C F1 D1	1197	JMP	STERR	
CEAB:	68	1198	IMPEX	PLA	
CEAC:	24 08	1199	RESPERR	BIT	ERRFLG
CEAE:	10 05	1200	BPL	DOREENT	
CEB0:	A2 FE	1201	LDR	#FE	;Resposta errada
CEB2:	4C A4 E6	1202	JMP	HANDLER	
CEB5:	A9 17	1203	DOREENT	LDA	NEEDIG
CEB7:	A0 00	1204	LDY	#NEEDIG	
CEB9:	20 5F CE	1205	JSR	STROUT	
CEBC:	A5 79	1206	LDA	OLOTEXT	
CEBE:	A4 7A	1207	LDY	OLOTEXT+1	
CEC0:	B5 B8	1208	STA	TXTPTR	
CEC2:	B4 B9	1209	STY	TXTPTR+1	
CEC4:	60	1210	RTS		
		1211			
CEC5:	20 20 06	1212	GET	JSR	ERRDIR

```

CECB: A2 01 1213      LDX R0H+1      ;Simular input
CECA: AD 02 1214      LDY R1H+1
CECC: AF 00 1215      LDA #000
CECE: 80 D1 02 1216      STA IN+1
CED1: AF 40 1217      LDA #540      ;Ligar INPUTFLG
CED3: 20 10 CF 1218      JSR MAINIMP
CED6: 60 1219      RTS
      1220
CED7: C9 22 1221 INPUT  CMP B***      ;Testar se veio string de
CED9: D0 0E 1222      BNE 000T      ;pedido de input
CEDB: 20 AF D1 1223      JSR STRTXT
CEDE: AF 30 1224      LDA R';'
CEED: 20 EB D1 1225      JSR SYNGRM
CEED: 20 62 CE 1226      JSR STRFRT
CEEA: 4C EC CE 1227      JMP 01R?
CEEY: 20 7F CE 1228      JSR OUTQUES      ;Nao veio string, imprimir??
CEEC: 20 20 D6 1229      JSR 000DIR
CEEF: AF 2C 1230      LDA H';'
DEF1: 80 FF D1 1231      STA IN-1
DEF4: 20 18 C8 1232      JSR INLIN
DEF7: AD D0 D2 1233      LDA IN
CEFA: C9 D3 1234      CMP #503      ;CONTROL-C?
CEFC: D0 1D 1235      BNE ZF
CEFE: 4C B8 C8 1236      JMP ERFLG?
      1237
CFD1: 20 7F CE 1238 HXIN  JSR OUTQUES
CFD4: 4C 18 C8 1239      JMP INLIN
CFD7: A6 7D 1240 READ  LDX DATPTR
CFD9: A4 7E 1241      LDY DATPTR+1
CFDB: AF 98 1242      LDA #590
CFDE: 2C 1243      HEX 2C
CFDE: AF 00 1244 ZF  LDA #500
CFD8: 85 15 1245 MAINIMP STA INPUTFLG
CFD2: B6 7F 1246      STX IMPTR
CFD4: B4 80 1247      STY IMPTR+1
CFD1A: 20 03 D3 1248 HXIMP JSR PTRGET
CFD19: B5 05 1249      STA FORPNT
CFD1B: B4 B6 1250      STY FORPNT+1
CFD1D: A5 B8 1251      LDA TXTPTR
CFD1F: A4 B9 1252      LDY TXTPTR+1
CFD21: B5 07 1253      STA TSPSV
CFD23: B4 B8 1254      STY TSPSV+1
CFD25: A6 7F 1255      LDX IMPTR
CFD27: A4 80 1256      LDY IMPTR+1
CFD29: B6 B8 1257      STX TXTPTR
CFD2B: B4 B9 1258      STY TXTPTR+1
CFD2D: 20 B7 D0 1259      JSR CRISGT
CFD30: D0 21 1260      BNE INSTART
CFD32: 24 15 1261      BIT INPUTFLG
CFD34: 50 11 1262      BVC 0000?      ;Desviar se nao for GET
CFD36: 20 18 FD 1263      JSR B44      ;Caso GET
CFD3F: 20 02 FE 1264      JSR IN2
CFD3C: 29 7F 1265      AND #57F
CFD3E: B0 D0 D2 1266      STA IN
CFD41: A2 FF 1267      LDX R0H-1
CFD43: AD D1 1268      LDY R1H-1
CFD45: D0 08 1269      BNE STOP

```

		1270		
CF4F:	30 7F	1271	SHOW?	BRJ FINDATA
CF4F:	20 7F CE	1272		JSR OUTWDS
CF4C:	20 01 CF	1273		JSR HXIN
CF4F:	04 00	1274	STX	TXTPTR
CF54:	04 09	1275	STY	TXTPTR+1
CF53:	20 01 00	1276	INSTART	JSR CHRGET
CF54:	24 11	1277	BIT	VALTYP
CF58:	10 31	1278	RPL	MUMIN
CF5A:	24 13	1279	BIT	INPUTFLS
CF5C:	50 09	1280	BVC	PUTCHR
CF5E:	E8	1281		DNX
CF5F:	04 00	1282	STX	TXTPTR
CF61:	A9 00	1283	LDA	#500
CF63:	05 00	1284	STA	CHARAC
CF65:	F0 0C	1285	BEQ	PENCHR
CF67:	05 00	1286	PUTCHR	STA CHARAC
CF69:	C9 22	1287	CMF	"' ' "
CF6B:	F0 07	1288	BEQ	PECHR
CF6D:	A9 3A	1289	LDA	"' ' "
CF6F:	05 00	1290	STA	CHARAC
CF71:	A9 2C	1291	LDA	"' ' "
CF73:	5B	1292	PENCHR	CLC
CF74:	05 0E	1293	PECHR	STA ENDCR
CF76:	A5 00	1294	LDA	TXTPTR
CF78:	A4 09	1295	LDY	TXTPTR+1
CF7A:	0F 00	1296	ADC	#500
CF7C:	F0 01	1297	BCC	SKP
CF7E:	C8	1298	DNX	
CF7F:	20 07 07	1299	SKP	JSR STBLT2
CF82:	20 57 0A	1300	JSR	POINT
CF85:	20 A0 C0	1301	JSR	PUTSTR
CF8B:	4C 9A CF	1302	JMP	MXE
CF8D:	4B	1303	MUMIN	PNA
CF8C:	A0 00 02	1304	LDA	IN
CF8F:	F0 30	1305	BEQ	INPFIN
CF91:	4B	1306	DATIN	PLA
CF92:	20 44 DF	1307	JSR	FEN
CF95:	A5 12	1308	LDA	INTFLG
CF97:	20 0B C0	1309	JSR	LET2
CF9A:	20 07 00	1310	MXE	JSR CHRGET
CF9D:	F0 07	1311	BEQ	SWPNT
CF9F:	C9 2C	1312	CMF	"' ' "
CA01:	F0 03	1313	BEQ	SWPNT
CA03:	4C 9A CE	1314	JMP	INPUTERR
		1315		
CA04:	A5 00	1316	SWPNT	LDA TXTPTR
CA0B:	A4 09	1317	LDY	TXTPTR+1
CA0A:	05 7F	1318	STA	SWPTR
CA0C:	04 00	1319	STY	SWPTR+1
CA0E:	A5 07	1320	LDA	TAPSV
CA0D:	A4 00	1321	LDY	TAPSV+1
CA02:	05 00	1322	STA	TXTPTR
CA04:	04 09	1323	STY	TXTPTR+1
CA06:	20 07 00	1324	JSR	CHRGET
CA0F:	F0 30	1325	BEQ	IMPONE
CA0E:	20 EA 01	1326	JSR	CHRCOM

Desviar se nao for GET

Saltar apostrofes se houver

Via de DATA?

Sim, desviar

Colocar numero FF em TXTPNT

Colocar em VWR1

Desviar se input terminado?
? ", "no input?"

Erro ao ler

Se o comand nao for executado,
o progr.deve ter uma virgula

```

CFB2: 4C 16 CF 1327      JNP NDIRP      ;Obter proximo input
CFC1: A5 15 1328 IMPFIN  LDA INPUTFLG
CFC3: 00 CC 1329      BNE DATIN
CFC5: 4C A6 CE 1330      JNP IMPERR
1331
CFCB: 20 CB CC 1332 FIMDATA JSR DATAN      ;Obter desLoc.do prox.comando
CFCB: CB 1333      DNY
CFCC: AA 1334      TAX
CFC0: 00 12 1335      BNE NXS      ;Fim da linha?
CFCF: A2 24 1336      LDX BMAADATA-NEVERBRO
CFD1: CB 1337      DNY
CFD2: B1 B8 1338      LDA (TXTPTR),Y ;Fim de programa?
CFD4: F0 5F 1339      BEB BERR      ;Sintax
CFD6: CB 1340      DNY
CFD7: B1 B8 1341      LDA (TXTPTR),Y ;Obter prox.numero de linha
CFD9: B5 78 1342      STA DATLIN
CFDB: CB 1343      DNY
CFDC: B1 B8 1344      LDA (TXTPTR),Y
CFDE: CB 1345      DNY
CFDF: B5 7C 1346      STA DATLIN+1
CFE1: B1 B8 1347 NXS     LDA (TXTPTR),Y ;Obter prim. token do comando
CFE3: AA 1348      TAX
CFE4: 20 B0 CC 1349      JSR ADDON      ;Atualizar TXTPTR
CFE7: E0 B3 1350      CFI Bdata
CFE9: 00 00 1351      BNE FIMDATA ;Loop ate encontrar DATA
CFEE: 4C 53 CF 1352      JNP INSTANT ;Token DATA encontrado
CFEE: A5 7F 1353 IMPONE  LDA IMPTR      ;Nao precisa mais de input
CFF0: A4 B0 1354      LOY IMPTR+1
CFF2: A6 15 1355      LDX INPUTFLG
CFF4: 10 03 1356      BPL NTO
CFF6: 4C 73 CB 1357      JNP SETDA      ;Se veio de DATA
CFF9: A0 00 1358 NTO     LOY B000
CFFB: B1 7F 1359      LDA (IMPTR),Y ;Input a mais?
CFFD: F0 07 1360      BEB RET11
CFFF: A9 07 1361      LDA NEXIG      ;Sim erro
D001: A0 00 1362      LOY R1EXIG
D003: 4C 5F CE 1363      JNP STROUT
D006: 60 1364 RET11     RTS
1323      PUT PARTESH,D1
11
D007: 45 58 54 12      EXIG     ASC 'EXTRA IGNORADO'0000
D00A: 52 41 20 49 47 4E 4F 52
D012: 41 44 4F 00 00
13
D017: 52 45 44 14      REDIG     ASC 'REDIGETE'0000
D01A: 49 47 49 54 45 00 00
15
D021: 00 04 16      NEXT     BNE VARNEXT ;Desviar se variavel
D023: A0 00 17      LOY B000      especificada
D025: F0 03 18      BEB SKPV
D027: 20 03 03 19      VARNEXT JSR PTRGET ;Achar pointer a variavel
D02A: 95 85 110      SKPV     STA FORPNT
D02C: 84 86 111      STY FORPNT+1
D02E: 20 54 C6 112      JSR GETFORPNT ;Achar dados de FOR
D031: F0 04 113      BEB GETFOR na pilha
D033: A2 00 114      LDX BMAwFOR-NEVERBRO
D035: F0 69 115      BEB JERROR ;Sempre

```

```

0037: 9A      116  007F0E  TXS           ;pontar a pilha
0038: E9      117      INX
0039: E9      118      INX
003A: E9      119      INX
003B: E9      120      INX
003C: 8A      121      T3A           ;Byte de baixa ordem
003D: E9      122      INX           ;do valor de STEP
003E: E9      123      INX
003F: E9      124      INX
0040: E9      125      INX
0041: E9      126      INX
0042: E9      127      INX
0043: 86 60    128      STX  DEST       ;Byte de baixa ordem
0045: A0 01    129      LDY  #S01       do endereco da variavel FOR
0047: 20 13 DE 130      JSR  NOVFN      ;STEP em FAC
004A: 8A      131      TSX
004B: 80 09 01 132      LDA  STACK+9,X
004E: 85 A2    133      STA  FACSEM
0050: A5 85    134      LDA  FORPNT
0052: A4 86    135      LDY  FORPNT+1
0054: 20 06 DA 136      JSR  FADD      ;Somar ao valor de FOR
0057: 20 41 DE 137      JSR  SETFOR   ;Devolver novo valor
005A: A0 01    138      LDY  #S01
005C: 20 CE DE 139      JSR  FCOMP2   ;Comparar com valor final
005F: 8A      140      TSX
0060: 38      141      SEC
0061: FD 09 01 142      SBC  STACK+9,X
0064: FD 17    143      BEQ  ENDFOR     ;Desviar se FOR terminado
0066: 80 DF 01 144      LDA  STACK+9F,X ;Mas, preparar numero
0069: 85 75    145      STA  CURLIN    da linha de FOR
006B: 80 10 01 146      LDA  STACK+$10,X
006E: 85 76    147      STA  CURLIN+1
0070: 80 12 01 148      LDA  STACK+$12,X;E posicionar TXTPTR logo apos
0073: 85 88    149      STA  TXTPTR    instracao FOR
0075: 80 11 01 150      LDA  STACK+$11,X
0078: 85 89    151      STA  TXTPTR+1
007A: 4C EC CA 152      GOMEST  JMP  NEWSTT
007D: 8A      153      ENDFOR  T3A
007E: 49 11    154      ADC  #S11      ;Liger CARRY
0080: A6      155      TAX           ;Cancelar FOR subindo
0081: 9A      156      TXS           ;a pilha de $12
0082: 20 87 00 157      JSR  CHR$01
0085: C9 2C    158      CMP  #' '      ;Outra variavel no NEXT?
0087: D0 F1    159      BNE  GOMEST
0089: 20 81 00 160      JSR  CHR$ET
008C: 20 27 00 161      JSR  WABCT     ;Mas retorna
162      *****
163      *
164      *      Avaliacao de formulas      *
165      *
166      *      Localizacao dos pointers    *
167      *
168      *      E tratamentos de strings    *
169      *
170      *****
008F: 20 A3 00 171      FIRMUM  JSR  FINEVL
0092: 18      172      CHONUM  CLC

```


0093:	24	173	HEX	24		
0094:	36	174	CHISTR	SEC		
0095:	24 11	175	CHVAL	BIT	VALTYP	
0097:	30 03	176	BNL	CV2		
0099:	80 03	177	BCS	ALSWCH		
009B:	80	178	RET12	RTS		
009C:	80 F3	179	CV2	NCS	RET12	
009E:	A2 84	180	RTSWCH	LOX	RTMCMF-HENSENRO	
00A0:	4C 01 C7	181	JERROR	JMP	ERROR	
		182				
		183			*Rotina principal de avaliacao de formulas	
		184			*Na entrada TXTPTR aponta ao primeiro caracter da formula	
		185				
00A3:	A6 80	186	FINVEVL	LOX	TXTPTR	
00A5:	00 02	187	BNE	ND1		
00A7:	C6 89	188	DEC	TXTPTR+1		
00A9:	C6 88	189	ND1	DEC	TXTPTR	
00AB:	A2 00	190	LOX	MS00		
00AD:	24	191	OFB	S24	;Prioridade inicial	
00AE:	48	192	FEVL00P	PHA	;Por na pilha ultima CPRTYP	
00AF:	8A	193	TBA			
00B0:	48	194	PHA		;E prioridade	
00B1:	A9 01	195	LOX	MS01		
00B3:	20 C5 C6	196	JSR	CHKMEN	;Testar a pilha != \$38	
00B6:	20 B8 01	197	JSR	GETVAL	;Obter valor do descritor	
00B9:	A9 00	198	LOX	MS00	de string em TXTPTR	
00BB:	85 89	199	STA	CPRTYP		
00BD:	20 B7 00	1000	FINVEVL2	JSR	CHKGET	
00CD:	38	1001	CPROP	SEC		
00CE:	E9 CF	1002	SBC	MS0F	;Token)	
00D0:	90 17	1003	BCC	CHRTYP		
00D5:	C9 03	1004	CMF	MS03	;ou =, (
00D7:	80 13	1005	BCS	CHRTYP	;Mas, desviar	
00D9:	C9 01	1006	CMF	MS01		
00DB:	2A	1007	NDL			
00DC:	49 01	1008	EDR	MS01		
00DE:	45 89	1009	EDR	CPRTYP	;Ligar bits de CPRTYP=00000 !=(
00E0:	C5 89	1010	CMF	CPRTYP		
00E2:	90 41	1011	BCC	SWTERR		
00E4:	85 89	1012	STA	CPRTYP		
00E6:	20 B1 00	1013	JSR	CHKGET	;Outro operador?	
00E9:	4C C3 00	1014	JMP	CPROP	;Testar de novo (,=,)	
00EC:	A6 89	1015	CHRTYP	LOX	CPRTYP	
00EE:	80 2C	1016	BNE	COMPARE	;Desviar se (,=,)	
00ED:	80 78	1017	BCS	NOTMATH	;Desviar se proximo token)"("&	
00E2:	49 07	1018	ADC	MS0F-plus		
00E4:	90 77	1019	BCC	NOTMATH	;Desviar se proximo token)"("&	
00E6:	45 11	1020	ADC	VALTYP	;"+ e ultimo resultado string?	
00E8:	80 03	1021	BNE	ARITH	;Mas, desviar	
00EA:	4C B1 00	1022	JMP	CAT	;Sim concatenar	
00ED:	69 FF	1023	ARITH	ADC	MSFF	;A contem deslocamento do "4"
00EF:	85 SE	1024	STA	INDEX		
00F1:	0A	1025	NDL			
00F2:	A5 SE	1026	ADC	INDEX	;Veze 3	
00F4:	A8	1027	TAT			
00F5:	A8	1028	PROFTST	PLA	;Obter ultima prioridade	
00F6:	89 92 C3	1029	CMF	MATHNBL_Y	;Comparar com priorid. corrente	

00F9: 80 67	1130	BDS	DMTH	;Se priorid. correta, executar
00FE: 20 92 D0	1131	JSR	CHNUM	;0 ultimo result. foi numero?
00FE: 48	1132	PNA		
00FF: 20 25 D1	1133	SAVOP	JSR	PSHMO
0100: 48	1134	PLA		;Guardar operacao na pilha
0103: A4 87	1135	LDY	LASTOP	
0105: 10 67	1136	BPL	PREFMC	;Desviar se ha mais formula
0107: NA	1137	TAX		
0108: F0 56	1138	BEQ	DMEX	;Se oper. salta, na forma. sair
010A: D0 5F	1139	BNE	DMATH	;Executar ultima operacao
	1140			
010C: 46 11	1141	COMPARE	LSR	VALTYP
010E: BA	1142	TXA		;Fazer CPRTYP=0000 != C com C=
010F: 2A	1143	ROL		;carry do ultimo
0110: A6 88	1144	LDX	TXTPTR	teste de caracteres
0112: D0 02	1145	BNE	ND2	
0114: CA 89	1146	DEC	TXTPTR+1	
0116: CA 88	1147	ND2	DEC	TXTPTR
0118: F0 18	1148	LDY	#PLUS-MATHTEL	;Forçar uso de POSOP para
011A: 85 89	1149	STA	CPRTYP	as 3 comparacoes
011C: D0 07	1150	BNE	PREFTEST	;Sempre
	1151			
011E: D9 92 C3	1152	PREFMC	CMP	MATHTEL,Y
0121: 80 48	1153	BDS	DMATH	;Executar se prioridade correta
0123: 90 09	1154	BCC	NDOP	;Obter mais formula
	1155			
0125: 89 94 C3	1156	PSHMO	LDA	MATHTEL+2,Y
0128: 48	1157	PNA		;Colocar endereco da rotina
0129: 89 93 C3	1158	LDA	MATHTEL+1,Y	automatica na pilha
012C: 48	1159	PNA		
0130: 20 38 D1	1160	JSR	PSHF	;Retornar na JMP(INDEX)
0130: A5 89	1161	LDA	CPRTYP	
0132: AC AE D0	1162	JMP	FEVLOOP	
	1163			
0135: 4C F1 D1	1164	SWTERR	JMP	SYNERR
	1165			
0138: A5 A2	1166	PSHF	LDA	FACSEN
013A: BE 92 C3	1167	LDX	MATHTEL,Y	;Obter FACSEN para coloca-lo
	1168			na pilha
	1169			
	116F			;Preparar INDEX para return e colocar FAC na pilha
	1170			; A contem FACSEN ou -1,0,1 se vier de STEP
	1171			
0130: A6	1172	PSHFAC	TAY	;Chamada por STEP
013E: 68	1173	PLA		;Tirar da pilha ender. de volta
013F: 85 5E	1174	STA	INDEX	;Coloca-lo em INDEX
0141: E6 5E	1175	INC	INDEX	;Esta rotina segue que o ender.
0143: A8	1176	PLA		;de volta nao esta em limite
0144: 85 5F	1177	STA	INDEX+1	de pagina
0146: 98	1178	TIA		
0147: 48	1179	PNA		;Colocar FACSEN na pilha
0148: 20 8C DE	1180	PUSHFAC	JSR	END
0148: A5 A1	1181	LDA	FAC+4	chamado por FOR
014D: 48	1182	PNA		
014E: A5 A0	1183	LDA	FAC+3	
0150: 48	1184	PNA		
0151: A5 9F	1185	LDA	FAC+2	
0153: 48	1186	PNA		

```

0154: A5 9E 187 LDA FAC+1
0156: 48 188 PMA
0157: A5 99 189 LDA FAC
0159: 48 190 PMA
015A: AC 5E 00 191 JMP (INDEX) ;Equivalente a RTS
                                324 PUT PARTE11,01
                                31
015D: A0 FF 12 NOTNATH LDY #5FF ;preparar saida
015F: 60 13 PLA
0160: F0 23 14 BCCX BEB EXIT ;Sem funcao matem a exec., sair
0162: C9 64 15 DONTM CMP #64 ;Era (=)?
0164: F0 03 16 BEB DINTM ;Sim, permitir compar. de string
0166: 20 92 00 17 JSR CNUMM
0167: 84 87 18 DINTM STY LASTOP
                                39
                                110 *Tirar numero ponto flutuante da pilha,
                                111 *colocar em ARG e ir a rotina matematica
                                112 *via RTS (o endereco foi colocado na pilha):
                                113 *(Observar que as rotinas ( = ) todas vao a F050F).
                                114
0168: 68 115 DONATH PLA
016C: 4A 116 LSR ;Restaurar carry
016D: 85 14 117 STA CPBMSK ;00000)=0
016F: 68 118 PLA
0170: 85 A5 119 STA ARG
0172: 68 120 PLA
0173: 85 A6 121 STA ARG+1
0175: 68 122 PLA
0176: 85 A7 123 STA ARG+2
0178: 68 124 PLA
0179: 85 A8 125 STA ARG+3
017B: 68 126 PLA
017C: 85 A9 127 STA ARG+4
017E: 68 128 PLA
017F: 85 AA 129 STA ARG+5
0181: 45 A2 130 EOR FACSGM
0183: 85 A8 131 STA SGNCPM
0185: A5 9D 132 EXIT LDA FAC ;Desviar a rotina com status
0187: 60 133 RTS ;posicionado por FAC
                                134
                                135 *Obter valor da variavel, funcao ou numero apos TXTPTR,
                                136 *ou apontar ao descritor de string se for string
                                137 *e coloca-lo em FAC. Esta rotina tambem avalia expressoes
                                138 *entre parenteses por chamada recursiva a FINEVL.
                                139 *E a rotina central de FINEVL.
                                140
018B: A7 00 141 GETVAL LDA #500
018A: 85 11 142 STA VALTYP
018C: 20 81 00 143 SKIP JSR CHGET
018F: 80 03 144 BCS VAR?
0191: 4C 64 0F 145 NUMBER JMP F1M ;Numerico
0194: 20 97 03 146 VAR? JSR ISLETC ;E variavel?
0197: 80 64 147 BCS UNIL
0199: C9 2E 148 CMP #'.'
019B: F0 F4 149 BEB NUMBER
019D: C9 C9 150 CMP #minus
019F: F0 55 151 BEB NEN

```

01A6: C9 C8	152	CMF	Bylus	
01A3: F0 E7	153	BEM	SKIP	
01A5: C9 22	154	CMF	B"m"	
01A7: 00 0F	155	BME	NOT?	
01A9: A5 00	156	STRTXT	LDA TXTPTR	String explicitos construir
01AB: A4 09	157	LDY	TXTPTR+1	descriptores
01AD: 49 00	158	ADC	#500	
01AF: 90 01	159	BCC	ST1	
01B1: C8	160	BNY		
01B2: 20 01 07	161	ST1	JSR STRLIT	
01B5: 4C 57 0A	162	JMP	POINT	Obter pointer a descrit em FAC
	163			
01B8: C9 C6	164	NOT?	CMF	Bnot
01BA: 00 10	165	BME	FN?	
01BC: A0 18	166	LDY	NUMOF-MATHTABL	
01BE: 00 30	167	BME	ENUL	Seempre
	168			
01C0: A5 90	169	ENULP	LDA FAC	Esta rotina so pode
01C2: 00 03	170	BME	NOT2	ser chamada por NOT atraves
01C4: A0 01	171	LDY	#501	do desvio anterior a ENUL
01C6: 2C	172	HEX	2C	
01C7: A0 00	173	NOT2	LDY	#500
01C9: 4C 18 06	174	JMP	SEMFLT	
	175			
01CC: C9 C2	176	FN?	CMF	#fa
01CE: 00 03	177	BME	SEM?	
01D0: 4C 6E 06	178	JMP	FUNCT	
	179			
01D3: C9 02	180	SEM?	CMF	Bsem
01D5: 90 03	181	BCC	PARCH	
01D7: 4C 34 02	182	JMP	UNARY	
	183			
01DA: 20 E3 01	184	PARCH	JSR CHROPH	Ha ou "/" em TXTPTR?
01DB: 20 A3 00	185	JSR	FINREV	Sim, avaliar e testar "/"
01ED: A9 29	186	CHICLS	LDA	B"")
01E2: 2C	187	HEX	2C	
01E3: A9 28	188	CHROPH	LDA	B"")
01E5: 2C	189	HEX	2C	
01E8: A9 2C	190	CHROPH	LDA	B", "
01EB: A0 00	191	SYNCH	LDY	#500
01EA: 01 00	192	CMF	(TXTPTR),Y	"/", em TXTPTR?
01EC: 00 03	193	BME	SYNCH	
01EE: 4C B1 00	194	JMP	CHIBET	Se OK, obter prox caract e ret
	195			
01F1: A2 DC	196	SYNCH	LDX	NSINTERRO-REMSERR
01F3: 4C 01 C7	197	JMP	ERROR	
	198			
01FA: A0 15	199	BNY	LDY	NUMOF-MATHTABL
01FB: 68	100	ENUL	PLA	
01F9: 68	101	PLA		
01FA: 4C FF 00	102	JMP	SAVDP	
	103			
01FD: 20 03 03	104	VARL	JSR PTRGET	
0200: 85 A0	105	STA	VPAT	
0202: F4 A1	106	STY	VPAT+1	
0204: A6 11	107	LDX	VALTYP	E string?
0206: F0 05	108	BEM	VER1	Naõ, desviar

0208:	A2 00	>109	LDX	#500	
020A:	86 AC	>110	STX	EXTRAFAC	
020C:	60	>111	RTS		
		>112			
020D:	A6 12	>113	VR1	LDX	INTFLG ;Variavel inteira ?
020F:	10 00	>114	BPL	VR2	;Nao, desviar
0211:	A0 00	>115	LDY	#500	
0213:	81 A0	>116	LDA	(VWNT),Y	;00ter byte de alta ordem
0215:	AA	>117	TAX		
0216:	D6	>118	INY		
0217:	81 A0	>119	LDA	(VWNT),Y	;00ter byte de baixa ordem
0219:	A6	>120	TAY		
021A:	9A	>121	TXA		
021B:	4C 0C 06	>122	JMP	DIWNTF	;Transformar em ponto flutuante
021E:	4C 13 0E	>123	VR2	JMP	NOUFA ;Mover (A,Y) a FAC
		>124			
0221:	20 A1 00	>125	SCREEN	JSR	CHRGET
0224:	20 40 E5	>126		JSR	PLOTFMS
0227:	BA	>127	TXA		
022B:	A4 FD	>128	LDY	FIRST	
022A:	20 6F FB	>129	JSR	SCRN	
022B:	A6	>130	TAY		
022E:	20 18 06	>131	JSR	SCANFLT	
0231:	4C ED 01	>132	JMP	CHRCLS	
		>133			
		>134	#Processar operadores unarios (funcoes)		
		>135			
0234:	C9 07	>136	UNARY	CMP	#cra ;Nao unario, especial
0236:	FD E9	>137	BEQ	SCREEN	
023B:	DA	>138	ASL		
023F:	48	>139	PMA		
023A:	AA	>140	TAX		
023B:	20 B1 00	>141	JSR	CHRGET	
023E:	ED CF	>142	CPX	01eflstra2-1	
0240:	90 20	>143	HCC	NOTinstr	;Devu se nao for oper de string
0242:	20 E3 01	>144	JSR	CHRCPM	;Testar "("
0245:	20 A3 00	>145	JSR	FRMEND	;processar concatenacao,...
024B:	20 E6 01	>146	JSR	CHRCOM	
024B:	20 94 00	>147	JSR	CHRSTR	;Certificar que e string
024E1:	68	>148	PLA		
024F:	AA	>149	TAX		;00ter pointer a rotina
0250:	A5 A1	>150	LDA	VWNT+1	
0252:	48	>151	PMA		
0253:	A5 A0	>152	LDA	VWNT	
0255:	48	>153	PMA		
0256:	BA	>154	TXA		
0257:	48	>155	PMA		;Coloca-o de volta na pilha
025B:	20 12 DA	>156	JSR	GETBYT	;00ter primeiro parametro em X
025B:	68	>157	PLA		
025C:	A6	>158	TAY		;Y apontando a rotina
025D:	BA	>159	TXA		
025E:	40	>160	PMA		;for primeiro paramet na pilha
025F:	4C A7 02	>161	JMP	GOROUT	;Exec rotina de string requer
		>162			
0262:	20 DA 01	>163	NOTinstr	JSR	PARDIR ;Testar sintaxe e avaliar arg.
0265:	68	>164	PLA		;00ter token n2
0266:	A0	>165	TAY		

```

0267: 89 82 C2 >166 GOROUT LDA UNFNC-S44,Y;S44 = sign2
026A: 85 91 >167 STA JMPADR5+1
026C: 89 83 C2 >168 LDA UNFNC-S43,Y
026F: 85 92 >169 STA JMPADR5+2
0271: 20 90 00 >170 JSR JMPADR5 ;Não retorna para LEFTS, NDS6,
0274: 4C 92 00 >171 JMP CHORUM RIGHTS
>172
0277: A5 A5 >173 OR LDA ARG
0279: 05 90 >174 ORA FAC
027B: 00 00 >175 BNE TRUE
027D: A5 A5 >176 AND LDA ARG
027F: F0 04 >177 BEQ FALSE
0281: A5 90 >178 LDA FAC
0283: 00 03 >179 BNE TRUE
0285: A0 00 >180 FALSE LDY #00
0287: 2C >181 HEX 2C
0289: A0 01 >182 TRUE LDY #01
028B: 4C 18 06 >183 JMP SCWFLT
>325
>326 PUT PARTE1,J,01
>327
>328 ;Retorna com as comparações (,,)
>329
028D: 20 95 00 >330 POPOP JSR CHEVAL
0290: 80 13 >331 RCS STRCMP ;Desviar se strings
0292: A5 A6 >332 LDA ARG50H ;Se ARG50H positivo, eliminar
0294: 09 7F >333 ORA #57F bit de alta ordem de ARG5
0296: 25 A6 >334 AND ARG+1
0298: 85 A6 >335 STA ARG+1
029A: A9 A5 >336 LDA ARG
029C: A0 00 >337 LDY #00
029E: 20 CC 0E >338 JSR FCOMP ;Retornar A= -1,0,1 conforme
02A1: AA >339 TAX ;ARG (,,) FAC
02A2: 4C 06 02 >340 JMP HUNCH
02A5: A9 00 >341 STRCMP LDA #00
02A7: 85 11 >342 STA VALTYP
02A9: C6 09 >343 DEC CTRTYP
02AB: 20 1A D9 >344 JSR FREEFAC
02AE: 85 90 >345 STA FAC ;Tamanho do string
02B0: 84 9E >346 STX FAC+1
02B2: 84 9F >347 STY FAC+2
02B4: A5 A0 >348 LDA ARG+3
02B6: AA A9 >349 LBT ARG+4
02B8: 20 1E 09 >350 JSR FRETHP
02BA: 84 A0 >351 STX ARG+3
02BC: 84 A9 >352 STY ARG+4
02BE: AA >353 TAX ;Tamanho do string ARG
02C0: 38 >354 SEC
02C1: E5 90 >355 SRC FAC ;Fazer X= menor tamanho
02C3: F0 00 >356 BNE SFS
02C5: A9 01 >357 LDA #01
02C7: 90 04 >358 BCC SFS
02C9: A6 90 >359 LDX FAC
02CB: A9 FF >360 LDA #FFF
02CD: 85 A2 >361 SFS STA FAC50H ;Indicar o mais curto
02CF: A0 FF >362 LDY #FFF
02D1: E8 >363 INX
02D2: C8 >364 CLOOP DRY

```

0203:	CA	139		DEX	
0204:	00 D7	140		BNE DDCMP	
0206:	A6 A2	141		LDX FACSGH	Se iguais ate agora, decidir
0208:	30 0F	142	MUNCH	BNE CYPONE	pelo tamanho
020A:	18	143		CLC	
020B:	90 0C	144		BCC CYPONE	
020E:	81 A8	145	DDCMP	LDA (ARG+3),Y	
020F:	01 9E	146		CMP (FAC+1),Y	
0211:	F0 07	147		BEQ CNFLOOP	
0213:	A2 FF	148		LDX BSYF	
0215:	80 02	149		BCC CYPONE	
0217:	A2 01	150		LDX #A01	
0219:	EB	151	CYPONE	INX	Converter FF,0,1 A 1,2,4
021A:	0A	152		TXA	
021B:	2A	153		ROL	
021C:	25 16	154		AND CPMASK	00000 >=0
021E:	F0 02	155		BEQ JF	Se nao bate, falso
021F:	A9 01	156		LDA #B01	Pelo menos um bate, verdadeiro
0221:	4C A0 0E	157	JF	JMP FLMT	
0225:	60	158		RTS	
		159			
02FA:	20 EA 01	160	KODIN	JSR CHKCOM	
02FB:	AA	161	GIN	TAX	
02FA:	20 08 03	162		JSR PTRGET2	criar e zerar arrays
02FD:	20 87 00	163		JSR CHRGOT	
0300:	90 F4	164		BNE KODIN	
0302:	60	165		RTS	
		166			
0303:	A2 00	167	PTRGET	LDX #B00	
0305:	20 87 00	168		JSR CHRGOT	Obter nome de variavel
0308:	06 10	169	PTRGET2	STX 01NFLG	X e nome da var.se vem de GIN
030A:	85 81	170	PTRGET3	STA VARNAME	Ponto de entrada de FNC
030C:	20 87 00	171		JSR CHRGOT	
030F:	20 97 03	172		JSR ISLETC	E uma letra?
0312:	80 03	173		BCC NAMOK	Sim, desviar
0314:	4C F1 01	174	NAMOK	JMP SYMERR	Nao, erro
0317:	A2 00	175	NAMOK	LDX #B00	
0319:	86 11	176		STX VALTYP	
031B:	86 12	177		STX INTFLG	
031D:	4C 21 03	178		JMP NORMAN	
0320:	00	179		BRE	
0321:	20 81 00	180	NORMAN	JSR CHRGOT	Segundo caract do nome da var.
0324:	90 05	181		BCC GTLT	Desviar se numerico
0326:	20 97 03	182		JSR ISLETC	E alfa?
0329:	90 08	183		BCC STRNG?	Nao, desviar
032B:	AA	184	GTLT	TAX	Guard seg. caract do nome em X
032C:	20 81 00	185	BYPASS	JSR CHRGOT	Achar end do nome da variavel
032F:	90 F8	186		BCC BYPASS	Loop se numerico
0331:	20 97 03	187		JSR ISLETC	
0334:	80 F6	188		BCC BYPASS	Ou alfa
0336:	C9 24	189	STRNG?	CMP #'S'	Ligar indicadores de tipo de variaveis
0338:	00 D6	190		BNE INTVAR?	
033A:	A9 FF	191		LDA BSYF	
033C:	85 11	192		STA VALTYP	Indicar string
033E:	00 10	193		BNE KIN	Sempre
0340:	C9 25	194	INTVAR?	CMP #'Z'	
0342:	00 13	195		BNE SCODH	

0344:	A5 14	196	LDA	SURFLS	;Variavel inteira permitida?
0346:	30 CC	197	BRI	8ADMAN	;Nao, erro
0348:	A9 00	198	LDA	MS00	
034A:	85 12	199	STA	INTFLG	;Indicar inteiro
034C:	05 B1	1100	ORA	VARMAN	
034E:	85 01	1101	STA	VARMAN	;Ligar bit de alta orden do
0350:	8A	1102	TXA		primeiro caracter do nome
0351:	09 00	1103	ORA	MS00	;Ligar bit de alta orden do
0353:	AA	1104	TAX		segundo caracter do nome
0354:	20 B1 00	1105	JSR	CHRGST	
0357:	86 82	1106	STX	VARMAN+1	;Segundo caracter do nome
0359:	38	1107	SEC		de variavel
035A:	05 14	1108	ORA	SURFLS	;Indices permitidos e array?
035C:	E7 20	1109	SEC	B'1'	
035E:	00 03	1110	BNE	BSB	;Nao, desviar
0360:	4C 38 DA	1111	JMP	ARRAY	
0363:	24 14	1112	BSR	RTT	SURFLS
0365:	30 02	1113	BMI	USEARCH	;Desv se vem de FOR, DEF ou FN
0367:	70 F7	1114	BVS	JMPY	;Desv se chamada por SETARIFT
0369:	A9 00	1115	VSEARCH	LDA	MS00
036B:	85 14	1116	STA	SURFLS	
036D:	A5 69	1117	LDA	VARTAB	;Inicializar pointer a variavel
036F:	A6 6A	1118	LDX	VARTAB+1	
0371:	A0 00	1119	LDY	MS00	
0373:	86 9C	1120	MOVAR	STX	LONTR+1
0375:	85 98	1121	MOV1	STA	LONTR
0377:	E4 6C	1122	CPX	ARYTAB+1	;fim das variaveis simples?
0379:	00 DA	1123	BNE	HW2	;Nao, continuar
037B:	C5 68	1124	CMP	ARYTAB	
037D:	F0 22	1125	BEB	NOTFND	;Sim, criar uma?
037F:	A5 B1	1126	LDX	VARMAN	
0381:	01 98	1127	CMP	(LONTR),Y	
0383:	00 08	1128	BNE	NOTFR	;Desviar se nao for esta
0385:	A5 82	1129	LDA	VARMAN+1	
0387:	C8	1130	INY		
038B:	01 98	1131	CMP	(LONTR),Y	
038A:	F0 6C	1132	BEB	SETVWHT	;Desviar se encontrada
039C:	08	1133	DEY		
038D:	18	1134	NOTFR	CLC	
038E:	A5 98	1135	LDA	LONTR	
0390:	69 07	1136	ADC	MS07	
0392:	90 E1	1137	BCC	HW1	
0394:	CB	1138	INX		
0395:	00 DC	1139	BNE	MOVAR	;Sempre
		1140			
		1141			*Verificar se e letra de A a Z. Ligar carry se for,
		1142			*desligar caso contrario
		1143			
0397:	C9 41	1144	ISLETC	CMP	B'A'
0399:	90 05	1145	BCC	RTB1	
039B:	E9 58	1146	SEC	B'Z'+1	
039D:	38	1147	SEC		
039E:	E9 A5	1148	SEC	MS100-'Z'+51	;Subtr A original
03A0:	60	1149	RTM1	RTS	
		1150			
03A1:	68	1151	NOTFND	PLA	;Obter baixa ord do end de cham
03A2:	48	1152	PMA		;Voltar a pilha

03A3: C9 FC	1153	CMF	#5FC	;Chamado por VARI?
03A5: D0 DF	1154	BNE	NEWVAR	;Nao, desviar
03A7: BA	1155	TSX		
03A8: 80 02 01	1156	LDA	STACK+2,X	;Alter alta ord do endereco de
03AB: C9 D1	1157	CMF	#VARI+2	chamada de VARI?
03AD: 00 D7	1158	BNE	NEWVAR	;Nao, desviar
03AF: A9 BA	1159	LDA	#TWORR	;Se nao e atribuido, eliminar
03B1: A0 D3	1160	LDI	#TWORR	endereco das variaveis de
03B3: 60	1161	RTS		;maneira que volte valor 0
03B4: 00	1162	TWORR	RRR	
03B5: 00	1163	RRR		
	1164			
	1165			;Mover arrays para dar espaco a novas variaveis
	1166			
03B6: A5 68	1167	NEWVAR	LDA	ARYTAB
03B8: A4 6C	1168		LDY	ARYTAB+1
03BA: 85 98	1169		STA	LOWTR
03BC: 84 9C	1170		STY	LOWTR+1
03BE: A5 60	1171		LDA	STREND
03C0: A4 6E	1172		LDY	STREND+1
03C2: 85 96	1173		STA	HIGHTR
03C4: 84 97	1174		STY	HIGHTR+1
03C6: 18	1175		CLC	
03C7: 6F D7	1176	ADC	#5D7	;Preparar para mover 7 bytes
03C9: 70 D1	1177	BCC	NAV	
03CB: C8	1178	INY		
03CC: 85 94	1179	NAV	STA	HIGHOS
03CE: 84 95	1180		STY	HIGHOS+1
03D0: 20 82 CA	1181	JSR	BLTU	;Mover
03D3: A5 94	1182		LDA	HIGHOS
03D5: A4 95	1183		LDY	HIGHOS+1
03D7: C8	1184	INY		;BLTU posiciona este valor
03D8: 85 68	1185		STA	ARYTAB
03DA: 84 6C	1186		STY	ARYTAB+1
03DC: A0 00	1187		LDY	#5D0
03DE: A5 81	1188		LDA	VARIAM
03E0: 91 98	1189		STA	(LOWTR),Y
03E2: C8	1190	INY		;Guardar nome de variavel
03E3: A5 82	1191		LDA	VARIAM+1
03E5: 91 98	1192		STA	(LOWTR),Y
03E7: A9 00	1193		LDA	#5D0
03E9: C8	1194	INY		;Colocar valor em 0
03EA: 91 98	1195		STA	(LOWTR),Y
03EC: C8	1196	INY		
03ED: 91 98	1197		STA	(LOWTR),Y
03EF: C8	1198	INY		
03F0: 91 98	1199		STA	(LOWTR),Y
03F2: C8	1200	INY		
03F3: 91 98	1201		STA	(LOWTR),Y
03F5: C8	1202	INY		
03F6: 91 98	1203		STA	(LOWTR),Y
03F8: A5 98	1204	SETVINT	LDA	LOWTR
03FA: 18	1205		CLC	
03FB: 6F 02	1206		ADC	#5D2
03FD: A4 9C	1207		LDY	LOWTR+1
03FF: 70 D1	1208	BCC	SUP	
0401: C8	1209	INY		

0402:	95 83	3210	SVP	STA	WARPNT	;Agora ao prim byte do valor
0404:	84 84	3211		STY	WARPNT+1	
0406:	60	3212		RTS		
		326		PUT	PARTES1,D1	
		31				
0407:	A5 0F	32	GETARY	LDA	MURDON	;Obter numero de dimensoes
0409:	0A	33	GETARY2	ASL		vezes 2
040A:	67 05	34		ADC	WPOS	;*(inome, desloc., num.de dim.)
040C:	A5 98	35		ADC	LWTR	;Somar ao pointer da variavel
040E:	A4 9C	36		LDY	LWTR+1	
0410:	90 01	37		BCC	GO	
0412:	C8	38		JNY		
0413:	85 94	39	GO	STA	ARYPNT	;Agora apontar ao primeiro
0415:	84 95	310		STY	ARYPNT+1	descriptor do array
0417:	60	311		RTS		
		312				
0418:	90 80 00	313	MEGRUN	HEX	90800000	; -32768
041B:	00					
		314				
041C:	20 81 00	315	NAKINT	JSR	CHRGCT	
041F:	20 8F 00	316		JSR	FNUMUN	
0422:	A5 A2	317	NRINT	LDA	FACSGN	;Erro se negativo
0424:	90 00	318		BMI	RT1	
0426:	A5 90	319	ATINT	LDA	FAC	
042B:	C9 90	320		CMP	WPO	;Nos(2*15)?
042A:	90 09	321		BCC	RT2	;Sim, desviar
042C:	A9 18	322		LDA	WMEGRUN	;= -2*15?
042E:	A0 04	323		LDY	B1MEGRUN	
0430:	20 CC 0E	324		JSR	FCORP	
0433:	00 7E	325	RT1	BNE	DRERR	;Nao, erro
0435:	4C 0C 0F	326	RT2	JMP	RTINT	
		327				
		328				
		329				
		330				
0438:	A5 14	331	NRBAT	LDA	SUBFLG	;Indices dados?
043A:	00 47	332		BNE	FINDART	;Nao, desviar
043C:	A5 10	333		LDA	DIMFLG	
043E:	05 12	334		ORA	INTFLG	;Ligar bit de alta ordem se 1
0440:	48	335		PHA		
0441:	A5 11	336		LDA	VALTYP	
0443:	48	337		PHA		
0444:	A0 03	338		LDY	WPO	
0446:	98	339	KXTD1H	TYA		
0447:	48	340		PHA		
0448:	A5 82	341		LDA	WMEGRUN+1	
044A:	48	342		PHA		
044B:	A5 81	343		LDA	WMEGRUN	
044D:	48	344		PHA		
044E:	20 1C 04	345		JSR	NAKINT	
0451:	68	346		PLA		
0452:	85 81	347		STA	WMEGRUN	
0454:	68	348		PLA		
0455:	85 82	349		STA	WMEGRUN+1	
0457:	68	350		PLA		
045B:	A8	351		TAY		
045F:	BA	352		TSX		

045A: 80 02 01 153	LDA	STACK+2,X	;Obter VALTYP e INTFLG
045B: 40 154	PWA		;e duplica-los
045E: 80 01 01 155	LDA	STACK+1,X	
0461: 40 156	PWA		
0462: A5 A0 157	LDA	FAC+3	;Obter indices e coloca-los na
0464: 90 02 01 158	STA	STACK+2,X	prilha no log de VALTYP e INTFLG
0467: A5 A1 159	LDA	FAC+4	
0469: 90 01 01 160	STA	STACK+1,X	
046C: C8 161	INY		
046D: 20 07 00 162	JSR	CHRGOT	
0470: C9 2C 163	CMF	#', '	
0472: FD 02 164	BEQ	KRTDIN	;Loop ate colocar todos os
0474: B4 0F 165	STY	HURDIN	indices na prilha
0476: 20 ED 06 166	JSR	CHCLS	
0479: 68 167	PLA		;Obter VALTYP e INTFLG
047A: 85 11 168	STA	VALTYP	
047C: 68 169	PLA		
047D: 85 12 170	STA	VALTYP+1	
047F: 29 7F 171	AND	B57F	;Mascarar bit de INTFLG
0481: 85 10 172	STA	DINFLG	obtido DINFLG
0483: A6 68 173	FMVARY	LDX	ARYTAB
0485: A5 4C 174	LDA	ARYTAB+1	
0487: B6 96 175	ARYLOOP	STX	LONTR
0489: 85 9C 176	STA	LONTR+1	
048B: C5 6E 177	CMF	STREND+1	
048D: D0 04 178	BNE	ARYTAB?	
048F: E4 6D 179	CFX	STREND	
0491: F0 3F 180	BEQ	NOTFOUND	
0493: A0 00 181	ARYTAB?	LDY	#\$00
0495: 81 98 182	LDA	(LONTR),Y	;Obter nome do array
0497: C8 183	INY		
049B: C5 E1 184	CMF	VARNAM	;É o procurado?
049A: D0 06 185	BNE	NAMEY	;Nao, desviar
049C: A5 02 186	LDA	VARNAM+1	
049E: D1 98 187	CMF	(LONTR),Y	
04A0: FD 16 188	BEQ	ARYFOUND	
04A2: C8 189	NAMEY	INY	
04A3: 81 98 190	LDA	(LONTR),Y	
04A5: 18 191	CLC		
04A6: A5 98 192	ADC	LONTR	
04A8: AA 193	TAX		
04A9: C8 194	INY		
04AB: 81 98 195	LDA	(LONTR),Y	
04AC: A5 9C 196	ADC	LONTR+1	
04AE: 99 D7 197	BCC	ARYLOOP	
04B0: A2 6F 198	SURERR	LDX	#NUNJILES+MEMSERIO
04B2: 2C 199	HEX	2C	
04B3: A2 34 100	INERR	LDX	#NUNJILES+MEMSERIO
04B5: 4C 01 C7 101	JER	JMP	ERROR
			102
04B8: A2 76 103	ARYFOUND	LDX	#NUNJILES+MEMSERIO
04BA: A5 10 104	LDA	DINFLG	
04BC: D0 F7 105	BNE	JER	
04BE: A5 14 106	LDA	SUMFLG	
04C0: FD 02 107	BEQ	CHRDIN	
04C2: 38 108	SEC		;Necessario para STORE
04C3: 40 109	RTS		;Saida caso veio de RETARYPT

0527:	B0 50	167	BCH	BNE	
0529:	B5 95	168	STA	ARYPNT+1	
052B:	A0	169	TAY		
052C:	BA	170	TXA		
052D:	A5 94	171	ADC	ARYPNT	
052F:	90 03	172	BCC	ZARY	
0531:	C8	173	INY		
0532:	F0 52	174	REB	BNE	
0534:	20 02 C6	175	ZARY	JSR	REASON ;Testar se ha espaço e
0537:	B5 60	176	STA	STREND	zerar o array
0539:	B4 6C	177	STY	STREND+1	
053B:	A9 00	178	LDA	#000	
053D:	E6 AE	179	INC	STRING2+1	
053F:	A4 60	180	LDT	STRING2	
0541:	F0 05	181	BEQ	NOPE	
0543:	80	182	ZLUP	DEY	
0544:	91 94	183	STA	(ARYPNT),Y	
0546:	00 FB	184	BNE	ZLUP	
054B:	C6 95	185	NOPE	DEC	ARYPNT+1 ;separar a proxima pagina
054A:	C6 AE	186	DEC	STRING2+1	
054C:	00 F5	187	BNE	ZLUP	;Loop ate terminar
054E:	E6 95	188	INC	ARYPNT+1	
0550:	38	189	SEC		
0551:	A5 60	190	LDA	STREND	;Computar deslocamento do
0553:	E5 98	191	SBC	LOWTR	proximo array
0555:	A0 02	192	LDT	#002	
0557:	91 98	193	STA	(LOWTR),Y	;Coloca-lo apos o nome
0559:	A5 6E	194	LDA	STREND+1	
055B:	C8	195	INY		
055C:	E5 9C	196	SBC	LOWTR+1	
055E:	91 98	197	STA	(LOWTR),Y	
0560:	A5 10	198	LDA	DIMFLG	;De DIM?
0562:	D0 62	199	BNE	RTM2	;Sim, desviar
	327		PUT	PARTESN,01	
0564:	C8	16	INY		
0565:	B1 98	17	FINDLEN	LDA	(LOWTR),Y ;Achar elemento especificado
0567:	B5 0F	18	STA	MURDIN	do array a partir de indice
0569:	A9 00	19	LDA	#000	colocado na pilha por NEXTDIN
056B:	B5 A0	15	STA	STRING2	
056D:	B5 AE	16	DIMLUP	STA	STRING2+1
056F:	C8	17	INY		
0570:	68	18	PLA		
0571:	A4	19	TAX		
0572:	B5 A0	110	STA	FAC+3	;Obter indice e testa-lo
0574:	68	111	PLA		contra dimensao
0575:	B5 A1	112	STA	FAC+4	
0577:	D1 98	113	CHF	(LOWTR),Y	
0579:	90 0E	114	BCC	DIMOK	
057B:	D0 D6	115	BNE	GSE	
057D:	C8	116	INY		
057E:	BA	117	TXA		
057F:	D1 98	118	CHF	(LOWTR),Y	
0581:	90 07	119	BCC	DIMOK2	
0583:	4C B0 D4	120	GSE	JMP	SUBERR
	121				
0586:	4C FF C6	122	BNE	JMP	MEMERR
	123				

0587:	C8	124	DINOK	IMY	
058A:	A5 AE	125	DINOK2	LDA	STRING2+1 ;Primeira vez?
058C:	05 AD	126		ORA	STRING2
058E:	18	127		CLC	
0591:	F8 DA	128		BEQ	KODM ;Sim, desviar
0591:	20 C7 05	129		JSR	MULT ;Calcular produto das dimensoes
0594:	8A	130		TXA	
0595:	A5 AD	131		ADC	FAC+3
0597:	AA	132		TAX	
0598:	98	133		TYA	
0599:	A4 SE	134		LDY	INDEX ;Obter Y guardado por MULT
059B:	A5 A1	135	KODM	ADC	FAC+4 ;Proximo DIN
059D:	86 AD	136		STX	STRING2
059F:	C6 DF	137		DEC	MUMDIN
05A1:	00 CA	138		BNE	DTHLUP ;Loop ate terminarem os indices
05A3:	85 AE	139		STA	STRING2+1
05A5:	A2 05	140		LDX	#500
05A7:	A5 B1	141		LDA	WRNMM
05A9:	10 D1	142		BPL	NINTA ;Desviar se nao for inteiro
05AB:	CA	143		DEX	
05AC:	A5 B2	144	NINTA	LDA	WRNMM+1
05AE:	10 02	145		BPL	FACT ;Desviar se real
05B0:	CA	146		DEX	
05B1:	CA	147		DEX	
05B2:	86 64	148	FACT	STX	RESULT+2
05B4:	A9 00	149		LDA	#500
05B6:	20 D0 05	150		JSR	MUL ;Multipl prod das dimens pelo
05B9:	8A	151		TXA	; tamanho de cada elemento
05BA:	A5 14	152		ADC	ARYPNT ;Somar endereco do array para
05BC:	85 B3	153		STA	WRYPNT ; obter endereco final
05BE:	98	154		TYA	
05BF:	A5 95	155		ADC	ARYPNT+1
05C1:	85 B4	156		STA	WRYPNT+1
05C3:	A8	157		TAY	
05C4:	A5 B3	158		LDA	WRYPNT
05C6:	60	159	RTN2	RTS	
		160			
		161			##ultiplicacao de 16 bits, ponto fixo, de (LOWTR),Y
		162			##com STRING2 com produto em A,X.
		163			##usada pelas rotinas de indexacao de arrays
		164			
05C7:	84 SE	165	MULT	STY	INDEX ;Guardar Y para recup apos RTS
05C9:	B1 98	166		LDA	(LOWTR),Y
05CB:	85 64	167		STA	RESULT+2
05CD:	88	168		DEY	
05CE:	B1 98	169		LDA	(LOWTR),Y
05D0:	85 65	170	MUL	STA	RESULT+3
05D2:	A9 10	171		LDA	#510 ;Indice para multipl 16 bits
05D4:	85 9F	172		STA	INDEX
05D6:	A2 00	173		LDX	#500
05D8:	A0 00	174		LDY	#500
05DA:	8A	175	MU2	TXA	;Deslocar X,Y a esquerda de 1 bit
05DB:	DA	176		ASL	
05DC:	AA	177		TAX	
05DE:	98	178		TYA	
05DF:	2A	179		ROL	
05E0:	A8	180		TAY	

05E0:	80 A4	181		BCC	ONE	;Erro se produto > 16 bits
05E2:	06 A0	182		ASL	STRNG2	;Deslocar para fora o bit
05E4:	26 AE	183		BOL	STRNG2+1	de alta ordem do multiplicador
05E6:	90 08	184		BCC	NU3	;Desviar se bit= 0
05E8:	18	185		CLC		
05E9:	0A	186		TXA		
05EA:	65 64	187		ADC	RESULT+2	;Soma pelo multiplicador a X,Y
05EC:	AA	188		TAX		
05ED:	98	189		TYA		
05EE:	65 65	190		ADC	RESULT+3	
05F0:	A8	191		TAY		
05F1:	80 93	192		BCC	ONE	;Erro se produto > 16 bits
05F3:	C6 99	193	NU3	DEC	INDX	
05F5:	00 E3	194		BNE	NU2	;Loop ate terminar
05F7:	60	195		RTS		
		196				
05F8:	A5 11	197	FRE	LDA	VALTYP	
05FA:	F0 03	198		BEQ	FRE2	
05FC:	20 1A 09	199		JSR	FREXAC	
05FE:	20 0E 07	100	FRE2	JSR	GABNAG	
0602:	38	1101		SEC		
0603:	A5 4F	1102		LDA	FRETOP	
0605:	E5 60	1103		SBC	STREND	
0607:	A8	1104		TAY		
0609:	A5 70	1105		LDA	FRETOP+1	
060A:	E5 6E	1106		SBC	STREND+1	
060C:	A2 00	1107	GIWYF	LOX	#500	;Transformar em ponto flutuante
		1108	*			po inteiro com sinal de A,Y
060E:	06 11	1109		STX	VALTYP	;indicar que e numero
0610:	85 9E	1110		STA	FAC+1	
0612:	84 9F	1111		STY	FAC+2	
0614:	A2 90	1112		LOX	#590	;Bicar DP 16 bits a direita
0616:	4C 85 0E	1113		JMP	FLD1	
0619:	A4 24	1114	POS	LDI	CH	
061B:	A9 00	1115	SGNFLT	LDA	#500	
061D:	38	1116		SEC		
061E:	F0 EC	1117		BEQ	GIWYF	
0620:	A6 76	1118	ERR0TB	LOX	CURLIN+1	
0622:	E8	1119		TRV		
0623:	00 A1	1120		BNE	RTW2	;Retornar se modo indireto
0625:	A2 96	1121		LOX	#CONDEG-HENSEBRO	
0627:	2C	1122		HEX	2C	
0628:	A2 00	1123	UNDFMC	LOX	#FUN+DEF-HENSEBRO	
062A:	4C 06 C7	1124		JMP	ERR0R	
		1125				
0629:	20 58 06	1126	DEF	JSR	FMC?	;Preparar variavel nome da funcao
0630:	20 20 06	1127		JSR	ERR0TB	
0633:	20 E3 01	1128		JSR	CHKOPN	
0636:	A9 00	1129		LDA	#580	
0638:	85 14	1130		STA	SUBFLG	;Nao permitir variaveis intoir.
063A:	20 03 03	1131		JSR	PTARGET	;Obter pointer aos argumentos
0639:	20 92 00	1132		JSR	CHKMUM	
0640:	20 ED 01	1133		JSR	CHKCLS	
0643:	A9 00	1134		LDA	Beval	
0645:	20 E8 01	1135		JSR	SYNEND	
0648:	48	1136		PAR		;Primeiro caracter apos "="
0649:	A5 04	1137		LDA	VARPNT+1	

064B:	4B	>138	PHA		
064C:	A5 B3	>139	LDA	VARPNT	
064D:	4B	>140	PHA		
064F:	A5 B9	>141	LDA	TXTPTR+1	
0651:	4B	>142	PHA		
0652:	A5 B8	>143	LDA	TXTPTR	
0654:	4B	>144	PHA		
0655:	20 BA CC	>145	JSR	DATA ;Saltar a proxima instrucao	
065B:	4C C9 D6	>146	JMP	FNCDATA ;Preparar pointers ao "valor"	
		>147			
		>148		*O "nome" de uma funcao e uma variavel simples	
		>149		*Caso nome tem a forma (neg,pos), seu "valor" contem	
		>150		* Pointer e defin	
		>151		* Pointer a variavel argumento	
		>152		* Primeiro caracter da definicao	
		>153			
065B:	A9 C2	>154	FNC?	LDA	Rfa
065D:	20 EB D1	>155	JSR	SYNCHW	
0660:	09 B0	>156	ORA	WSB3	
0662:	B5 14	>157	STA	SUBFLG	;Indicar variavel simples e lig
0664:	20 BA D3	>158	JSR	PTRGET3	bit de alta ordem do primeiro
		>159	*		caracter do nome
0667:	B5 BA	>160	STA	FNCNAM	;Guardar o pointer
0669:	B4 B8	>161	STY	FNCNAM+1	
066B:	4C 92 D0	>162	JMP	CHRMUN	
		>163			
066E:	20 58 D6	>164	FUNCT	JSR	FNC?
0671:	A5 B8	>165	LDA	FNCNAM+1	;Obter pointer ao nome da func.
0673:	4B	>166	PHA		
0674:	A5 BA	>167	LDA	FNCNAM	
0676:	4B	>168	PHA		
0677:	20 BA D1	>169	JSR	PARCHK	;Avaliar argumento (em FAC)
067A:	20 92 D0	>170	JSR	CHRMUN	
067D:	68	>171	PLA		
067E:	B5 BA	>172	STA	FNCNAM	
0680:	68	>173	PLA		
0681:	B5 B8	>174	STA	FNCNAM+1	
0683:	A0 D2	>175	LDY	WSB2	
0685:	B1 BA	>176	LDA	(FNCNAM),Y	;Obter pointer ao argumento
0687:	B5 B3	>177	STA	VARPNT	
0689:	AA	>178	TAX		
068A:	C8	>179	INY		
068B:	B1 BA	>180	LDA	(FNCNAM),Y	
068D:	F0 99	>181	BEQ	UNDFNC	;Se byte de alta ordem 0,
068F:	B5 B4	>182	STA	VARPNT+1	nao estava definido
0691:	C8	>183	INY		
0692:	B1 B3	>184	LDA	(VARPNT),Y	;Salvar valor da variavel
0694:	4B	>185	PHA		argumento
0695:	B8	>186	BEY		
0696:	10 FA	>187	BPL	SAVOLD	
069B:	AA BA	>188	LDY	VARPNT+1	;Apontar ao valor do argumento
069A:	20 45 DE	>189	JSR	MOVWF	;FAC -> (VARPNT)
069D:	A5 B9	>190	LDA	TXTPTR+1	;Guardar posicao
069F:	4B	>191	PHA		
06A0:	A5 B8	>192	LDA	TXTPTR	
06A2:	4B	>193	PHA		
06A3:	B1 BA	>194	LDA	(FNCNAM),Y	;Y= 0

06A5: 05 00	1195	STA	TXPTR	;Apontar a definicao da funcao
06A7: C0	1196	INY		
06A8: 01 0A	1197	LDA	(FNCNAM),Y	
06AA: 05 09	1198	STA	TXPTR+1	
06AC: A5 04	1199	LDA	VARPNT+1	
06AE: 40	1200	PLA		
06AF: A5 03	1201	LDA	VARPNT	
06B1: 40	1202	PLA		
06B2: 20 0F 00	1203	JSR	FNMAIN	;Avaliar a funcao
06B5: 60	1204	PLA		
06B6: 05 0A	1205	STA	FNCNAM	
06B8: 60	1206	PLA		
06B9: 05 0B	1207	STA	FNCNAM+1	
06BB: 20 07 00	1208	JSR	CHRGST	;Tem ou nao fim de instrucao
06BE: F0 03	1209	BEQ	GETOLD	
06C0: 4C F1 D1	1210	JMP	SWERR	
	1211	PUT	PARTEEM D1	
	1212			
06C3: 60	1213	GETOLD	PLA	;Obter posicao do programa
06C4: 05 0B	1214	STA	TXPTR	
06C6: 60	1215	PLA		
06C7: 05 09	1216	STA	TXPTR+1	
06C9: A0 00	1217	FNCDATA	LDY	#000 ;Obter valor da variavel argum
06CB: 60	1218	PLA		
06CC: 01 0A	1219	STA	(FNCNAM),Y	
06CE: 60	1220	PLA		
06CF: C0	1221	INY		
06D0: 01 0A	1222	STA	(FNCNAM),Y	
06D2: 60	1223	PLA		
06D3: C0	1224	INY		
06D4: 01 0A	1225	STA	(FNCNAM),Y	
06D6: 60	1226	PLA		
06D7: C0	1227	INY		
06D8: 01 0A	1228	STA	(FNCNAM),Y	
06DA: 60	1229	PLA		
06DB: C0	1230	INY		
06DC: 01 0A	1231	STA	(FNCNAM),Y	
06DE: 60	1232	RTS		
	1233			
06DF: 20 02 00	1234	STB	JSR	CHRGUM ;Certificar que e um numero
06E1: A0 00	1235	LDY	#000	
06E4: 20 50 E0	1236	JSR	FACSTRNG	;Converter a string na pilha
06E7: 60	1237	PLA		
06E8: 60	1238	PLA		
06E9: A9 FF	1239	LDA	#0FF	;Apontar a pilha -1 para forcar
06EB: A0 00	1240	LDY	#000	movimento da string
06ED: F0 12	1241	BEQ	STRLIT	;Criar descritor e mover string
	1242			
	1243			
	1244			
06EF: A6 A0	1245	STRNG	LDX	FAC+3
06F1: A4 A1	1246		LDY	FAC+4
06F3: B6 BC	1247		STX	DESCPTR
06F5: 04 80	1248		STY	DESCPTR+1
06F7: 20 6C 07	1249	STBSPA	JSR	GETSPA ;p tem o tamanho
06FA: B6 9E	1250		STX	FAC+1 ;Guardar descritor em FAC
06FC: 04 9F	1251		STY	FAC+2

06FE: 05 90	141	STA	FAC	
0700: 50	142	RTS		
	143			
0701: A2 22	144	STRLIT	LDR #'''	
0703: 06 00	145	STX	CHARAC	Pres delimitadores de literal
0705: 04 0E	146	STX	ENDCHR	
0707: 05 A0	147	STRLT2	STA STRING1	
0709: 04 AC	148	STY	STRING+1	
070B: 05 9E	149	STA	FAC+1	Para descritor
070D: 04 9F	150	STY	FAC+2	
070F: A0 FF	151	LDR	#FF	
0711: 0B	152	FEND	IMY	achar fim do string
0713: 01 AB	153	LDA	(STRING),Y	
0715: F0 0C	154	BEQ	ZEND	
0717: C5 00	155	CMF	CHARAC	
0719: F0 04	156	BEQ	NOV?	
071A: C5 0E	157	CMF	ENDCHR	
071C: 00 F3	158	BNE	FEND	
071E: C9 22	159	CMF	#'''	
0720: F0 01	160	BEQ	NZ	
0722: 10	161	ZEND	CLC	
0723: 04 90	162	NZ	STY	FAC
0725: 90	163	TBA		Tamanho no desc temporario
0726: 45 A0	164	ADC	STRING1	
0728: 05 A0	165	STA	STRING2	Apontar ao fim do string
072A: A6 AC	166	LDR	STRING+1	
072C: 90 01	167	RCC	FEL	
072E: EB	168	END		
072F: 06 AE	169	FEL	STX	STRING2+1
0731: 45 AC	170	LDA	STRING+1	
0733: F0 04	171	BEQ	FE2	String e movido, se esta em
0735: C9 02	172	CMF	#002	FFF ou no buffer de entrada
0737: 00 00	173	BNE	PUTHEW	Caso contr., so prep. descritor
0739: 90	174	FE2	TBA	Obter tamanho em A
073A: 20 EF 06	175	JSR	STRING	para espaco para string
073C: A6 A0	176	LDR	STRING1	
073E: A4 AC	177	LDR	STRING+1	
0741: 20 FC 08	178	JSR	MOVSTR	move-to
0744: A6 52	179	PUTHEW	LDR	TEMPPT
0746: ED 5E	180	CPX	TEMPST+9	Descritores temporarios demais?
0748: 00 05	181	BNE	PUTHEW	
074A: A2 F0	182	LDR	INFORMACC+HEXERR0	
074C: 4C 01 C7	183	JERR	JMP	ERR0
	184			
074F: 45 90	185	PUTHEW	LDA	FAC
0751: 95 00	186	STA	LOC0,X	Apontar descritor temporario
0753: 45 9E	187	LDA	FAC+1	
0755: 95 01	188	STA	LOC1,X	
0757: 45 9F	189	LDA	FAC+2	
0759: 95 02	190	STA	LOC2,X	
075B: A0 00	191	LDR	#000	
075D: 06 A0	192	STX	FAC+3	
075F: 04 A1	193	STY	FAC+4	
0761: 00	194	DEY		
0763: 04 11	195	STY	WALTYP	Indicar string
0765: 06 53	196	STX	LASTPT	Apontar ao proximo descritor
0767: EB	197	END		

0767:	ED	Y98	INX	
0768:	ED	Y99	INX	
0769:	86 52	X100	STX TEMPT	
076B:	6D	X101	RTS	
		X102		
		X103		*Criar espaço para string, tamanho em B
		X104		
076C:	46 13	X105	GETSPA	LSR GARFLG ;Permitir reorganizacao da area
076E:	48	X106	GETSPC	PHA
076F:	49 FF	X107	EDM	BSFF
0771:	38	X108	SEC	
0772:	65 6F	X109	ABC	FRETOP ;Subtrair tamanho de FRETOP
0774:	A4 7D	X110	LDY	FRETOP+1
0776:	8D D1	X111	BCL	CT
0778:	88	X112	DEY	
0779:	C4 6E	X113	CT	CPY STREND+1
077B:	9D 11	X114	BCC	FULL ;Desviar se nao houver espaço
077D:	0D D4	X115	BNE	GETSPA
077F:	C5 6D	X116	CMP	STREND
0781:	9D D8	X117	BCC	FULL
0783:	85 6F	X118	GETSPA	STA FRETOP
0785:	84 7D	X119	STY	FRETOP+1
0787:	85 71	X120	STA	FRETSPC
0789:	84 72	X121	STY	FRETSPC+1
078B:	AH	X122	TAX	
078C:	68	X123	PLA	
078D:	6D	X124	RTS	
		X125		
078E:	A2 4D	X126	FULL	LDX #54D
0790:	A5 63	X127	LDA	GARFLG ;Ja houve reorganizacao?
0792:	3D B8	X128	BNE	JERR ;Sim, erro
0794:	2D 9E 07	X129	JSR	GARBAG
0797:	89 8D	X130	LDA	#59D ;Indicar reorganizacao feita
0799:	85 13	X131	STA	GARFLG
079B:	68	X132	PLA	
079C:	0D D0	X133	BNE	GETSPC
079E:	A6 73	X134	GARBAG	LDX MEMSIZ ;Reorganizar a partir do comeco
07A0:	A5 74	X135	LDA	MEMSIZ+1
07A2:	86 6F	X136	FNDVAR	STX FRETOP ;Uma passada por todas as
07A4:	85 7D	X137	STA	FRETOP+1 ;variaveis para cada string ativo
07A6:	AD D0	X138	LDY	#50D
07A8:	84 68	X139	STY	FNDVAR+1 ;Ind.que nao houve ainda reorg.
		X140		
		X141		*Apontar LOWTR ao fim da area de string
		X142		
07AB:	A5 6D	X143	LDA	STREND
07AC:	A6 6E	X144	LDX	STREND+1
07AE:	85 96	X145	STA	LOWTR
07B0:	86 9C	X146	STX	LOWTR+1
07B2:	A9 55	X147	LDA	RTMPST ;Apontar a descritores de
07B4:	A2 D0	X148	LDX	#XTEMPST ;strings temporarios
07B6:	85 5E	X149	STA	INDEX
07B8:	86 5F	X150	STX	INDEX+1
07BA:	C5 52	X151	TWAI	CMP TEMPT ;Fim dos temporarios?
07BC:	F0 D5	X152	BNE	SWABS ;Sim, or a variaveis simples
07BE:	2D 3D D8	X153	JSR	OWAR ;Tratar um temporario
07C1:	F0 F7	X154	BNE	TWAI ;Sempre

		1155		
07C3:	A9 07	1156	SWMS	LDA #507
07C5:	85 8F	1157		STA OSCLEN
07C7:	A5 89	1158		LDA UNTAB
07C9:	A4 4A	1159		LDX UNTAB+1
07CB:	85 5E	1160		STA INDEX
07CD:	86 5F	1161		STX INDEX+1
07CF:	E4 AC	1162	SWAR	CPX ARYTAB+1 ;Fim de variaveis simples?
07D1:	00 04	1163	BNE SWARGO	;Nao, continuar
07D3:	C5 A8	1164	CMF ARYTAB	
07D5:	FD 05	1165	BEQ ARYVAR	;Sim, tratar arrays
07D7:	20 33 08	1166	SWARGO	JSR SWMS ;tratar variaveis simples
07DA:	FD F3	1167	BEQ SWAR	;Sempre
		1168		
07DC:	85 94	1169	ARYVAR	STA ARYPNT
07DE:	86 95	1170		STX ARYPNT+1
07E0:	A9 03	1171		LDA #503
07E2:	85 8F	1172		STA OSCLEN
07E4:	A5 94	1173	ARYV2	LDA ARYPNT
07E6:	A6 95	1174		LDX ARYPNT+1
07E8:	EA AE	1175	ARYV3	CPX STREND+1 ;Fim de arrays?
07EA:	00 07	1176	BNE ARYV0	;Nao, tratar um array
07EC:	C5 A0	1177	CMF STREND	
07EE:	00 03	1178	BNE ARYV0	
07F0:	4C 7C 80	1179	JMP GBEPAS	;Todas as variaveis testadas,
		1180	*	; mover a de cima
		329		
07F3:	85 5E	11	ARYV0	PUT PARTE10,01
07F5:	86 5F	12		STA INDEX
07F7:	A0 00	13		STX INDEX+1
07F9:	81 5E	14		LDY #500
07FB:	A0	15		LDA (INDEX),Y ;Obter nome do array
07FC:	C8	16		TAX
07FD:	81 5E	17		LDY (INDEX),Y
07FF:	08	18	PNP	;Guardar seu tipo
0800:	C8	19	JNY	
0801:	91 5E	110	LDA (INDEX),Y	;Obter desloc do proximo array
0803:	A5 94	111	ADC ARYPNT	;Computar enderecos
0805:	85 94	112	STA ARYPNT	;E apontar a eles
0807:	C8	113	JNY	
0808:	81 5E	114	LDA (INDEX),Y	
080A:	A5 95	115	ADC ARYPNT+1	
080C:	85 95	116	STA ARYPNT+1	
080E:	26	117	PLP	
080F:	10 03	118	BPL ARYV2	;Desviar se nao for string
0811:	8A	119	TXA	
0812:	20 00	120	BNI ARYV2	;Desviar se nao for string
0814:	C8	121	JNY	
0815:	91 5E	122	LDA (INDEX),Y	;Obter numero de dimensoes
0817:	A0 00	123	LDY #500	
0819:	0A	124	ASL	
081A:	A9 05	125	ADC #505	
081C:	A5 5E	126	ADC INDEX	;Apontar ao primeiro elemento
081E:	85 5E	127	STA INDEX	do array
0820:	90 02	128	BCC BPL1	
0822:	E6 5F	129	INC INDEX+1	
0824:	A6 5F	130	BPL1	LDX INDCX+1

0826:	E4	75	131	ARYSTR	CPX	ARYPNT+1	;Fim de array?
0828:	00	04	132	BNE	GOGO		;Nao, tratar proximo elemento
082A:	CC	74	133	CPY	ARYPNT		
082C:	FD	8A	134	BEQ	ARYV33		;Sim, proximo array
082E:	20	3D	08	135	GOGO	JSR	DVAR
0830:	FD	F3	136	BEQ	ARYSTR		;Sempre
			137				
0833:	81	5E	138	DWARS	LDA	(INDEX),Y	;Variavel int ou def de funcao?
0835:	30	35	139	BMI	DWARTS		;Sim, saltar
0837:	C8		140	INY			
0838:	81	5E	141	LDA	(INDEX),Y		;Variavel string?
083A:	10	3D	142	BPL	DWARTS		;Nao, saltar
083C:	C8		143	INY			
083D:	81	5E	144	DVAR	LDA	(INDEX),Y	;Obter tamanho
083F:	FD	28	145	BEQ	DWARTS		;Ignorar se tamanho 0
0841:	C8		146	INY			
0842:	81	5E	147	LDA	(INDEX),Y		;Obter endereco do string
0844:	AA		148	TAX			
0845:	C8		149	INY			
0846:	81	5E	150	LDA	(INDEX),Y		
0848:	CC	7D	151	CMF	FRETOP+1		
084A:	9D	04	152	BCC	DV1		
084C:	D0	1E	153	BNE	DWARTS		
084E:	E4	67	154	CPX	FRETOP		
0850:	8D	1A	155	BCS	DWARTS		;Saltar se ja reorganizado
0852:	CC	9C	156	DV1	CMF	LWTR+1	;Achado ate acima do ult.string?
0854:	9D	16	157	BCC	DWARTS		;Nao, saltar
0856:	D0	04	158	BNE	DV2		;Sim, apontar a este ponto
0858:	E4	98	159	CPX	LWTR		
085A:	9D	1D	160	BCC	DWARTS		
085C:	86	98	161	DV2	STX	LWTR	
085E:	85	9C	162	STA	LWTR+1		
0860:	A5	5E	163	LDM	INDEX		
0862:	A6	5F	164	LDM	INDEX+1		
0864:	85	0A	165	STA	FNCNAM		
0866:	86	8B	166	STX	FNCNAM+1		
0868:	A5	8F	167	LDM	OSOLEN		
086A:	85	91	168	STA	BLENGTH		
086C:	A5	8F	169	DWARTS	LDM	OSOLEN	;Preparar para proxima variavel
086E:	18		170	CLC			
086F:	65	5E	171	ADC	INDEX		
0871:	85	5E	172	STA	INDEX		
0873:	9D	02	173	BCC	VOONE		
0875:	E6	5F	174	INC	INDEX+1		
0877:	A6	5F	175	VOONE	LDM	INDEX+1	
0879:	A0	D0	176	LDM	NS00		
087B:	60		177	RTS			
			178				
			179				;Feita a passagem pelas variaveis, mover agora o ultimo
			180				string ao topo e voltar para tratar outro.
			181				
087C:	A6	8B	182	OROPAS	LDM	FNCNAM+1	;Reorganizacao feita?
087E:	FD	F7	183	BEQ	VOONE		;Sim, retornar
0880:	A5	91	184	LDM	BLENGTH		
0882:	29	04	185	AND	NS04		;4 se simples, 0 caso contrario
0884:	AA		186	LSR			
0885:	AB		187	TAY			

DB64: 85 91	>88	STA BLENGTH	;2 se simples, 0 caso contrario
DB68: 81 8A	>89	LDA (FNCNAN),Y	
DB6A: 65 98	>90	ADC LONTR	
DB6C: 85 96	>91	STA HIGHTR	
DB6E: A5 9C	>92	LDA LONTR+1	
DB70: A9 00	>93	ADC #500	
DB72: 85 97	>94	STA HIGHTR+1	
DB74: A5 A7	>95	LDA FRETOP	
DB76: A6 7D	>96	LDX FRETOP+1	
DB78: 85 94	>97	STA HIGHDS	
DB7A: 86 95	>98	STX HIGHDS+1	
DB7C: 2D 89 C6	>99	JSR BLTCL2	;Mover string para cima e
DB7F: A4 91	>100	LDY BLENGTH	fixar seu descritor
DBA1: C8	>101	INY	
DBA2: A5 94	>102	LDA HIGHDS	
DBA4: 91 8A	>103	STA (FNCNAN),Y	
DBA6: AA	>104	TAX	
DBA7: E6 95	>105	INC HIGHDS+1	
DBA9: A5 95	>106	LDA HIGHDS+1	
DBAB: C8	>107	INY	
DBAC: 91 8A	>108	STA (FNCNAN),Y	;3.A aponta agora ao string mov.
DBAE: AC A2 D7	>109	JMP FNDVAR	;Procurar outro para deslocar
	>110		
DBE1: A5 A1	>111	LDA FAC+4	;Salvar prim. pointer do descr.
DBE3: A8	>112	PNA	
DBE4: A5 A0	>113	LDA FAC+3	
DBE6: A8	>114	PNA	
DBE7: 2D 88 D5	>115	JSR GETVAL	
DBE9: 2D 94 D0	>116	JSR CHGSTR	;Obt.point.ao descr.do segundo
DBEB: 68	>117	PLA	;Recuperar primeiro
DBED: 85 A8	>118	STA STRNG1	pointer do descritor
DBEF: 68	>119	PLA	
DBF1: 85 AC	>120	STA STRNG1+1	
DBF3: A8 D0	>121	LDY #500	
DBF5: 81 A8	>122	LDA (STRNG1),Y	;Somar tamanhos
DBF7: 18	>123	CLC	
DBF9: 71 A0	>124	ADC (FAC+3),Y	
DBFB: 90 05	>125	BCC NTL	;OK se < \$100
DBFD: A2 80	>126	LDX NEXTSTR0-NEXTSTR0	
DBFF: 4C D1 C7	>127	JMP ERROR	
	>128		
DB01: 2D EF D4	>129	JSR STRNG1	;Obt.espaço p/ concatén. string
DB03: 2D EE D6	>130	JSR MOVIMS	;Mover primeiro string
DB05: A5 8C	>131	LDA DSCPTR	;Liberar o segundo
DB07: A4 80	>132	LDY DSCPTR+1	
DB09: 2D 1E D9	>133	JSR FRETRP	
DB0B: 2D 00 D9	>134	JSR MOVSTR	;Mover segundo string
DB0D: A5 A8	>135	LDA STRNG1	;Liberar o primeiro
DB0F: A4 AC	>136	LDY STRNG1+1	
DB11: 2D 1E D9	>137	JSR FRETRP	
DB13: 2D 44 D7	>138	JSR PUTNEW	;Preparar o descritor
DB15: AC 80 D0	>139	JMP FNDVAL2	;Mais formula
	>140		
DBED: A8 D0	>141	LDY #500	;Mover string cujo descritor
DBFD: 81 A8	>142	LDA (STRNG1),Y	esta em (STRNG1) para (FRESPC)
DBFF: A8	>143	PNA	;Tamanho
DB01: C8	>144	INY	

08F4:	81	A8	:145	LDA	(STRING1),Y		
08F5:	A6		:146	TAX		;Par pointer ao string em X,Y	
08F7:	C8		:147	DMY			
08F8:	81	A8	:148	LDA	(STRING1),Y		
08FA:	A8		:149	TAY			
08FB:	68		:150	PLA		;Obter tamanho	
08FC:	86	5E	:151	MOVSTR	STX INDEX	;Mover string a X,Y	
08FE:	84	5F	:152	STY	INDEX+1	;Em INDEX para (FRESPEC)	
0900:	A8		:153	MOVSTR	TAY	;Tamanho em A)	
0901:	F0	0A	:154	BEQ	MVS3		
0903:	48		:155	PHA			
0904:	88		:156	MVS2	DEY		
0905:	81	5E	:157	LDA	(INDEX),Y		
0907:	91	71	:158	STA	(FRESPEC),Y		
0909:	98		:159	TYA			
090A:	00	F8	:160	BNE	MVS2		
090C:	68		:161	PLA			
090D:	18		:162	MVS3	CLC		
090E:	65	71	:163	ADC	FRESPEC		
0910:	85	71	:164	STA	FRESPEC		
0912:	90	02	:165	BCC	BPL2		
0914:	E6	72	:166	INC	FRESPEC+1		
0916:	60		:167	BPL2	RTS		
			:168				
0917:	20	94	00	FRESTR	JSR	DIRSTR	;Ultimo resultado e string?
091A:	A5	A0	:170	FREFAC	LDA	FAC+3	;Obter pointer ao descritor
091C:	A4	A1	:171	LDT	FAC+4		
091E:	85	5E	:172	FRETMP	STA	INDEX	;liberar descritor temporario
0920:	84	5F	:173	STY	INDEX+1		cujo pointer esta em (A,Y)
0922:	20	4F	09	:174	JSR	FRETMS	;libera descritor se temporario
0925:	08		:175	PHP			;Guardar se ultimo liberado
0926:	A0	00	:176	LDT	#500		
0928:	81	5E	:177	LDA	(INDEX),Y		
092A:	48		:178	PHA			;Colocar tamanho na pilha
092B:	C8		:179	INY			
092C:	81	5E	:180	LDA	(INDEX),Y		
092E:	4A		:181	TAX			;Obter pointer ao string em X,Y
092F:	C8		:182	INY			
0930:	81	5E	:183	LDA	(INDEX),Y		
0932:	A8		:184	TAY			
0933:	68		:185	PLA			;Obter tamanho e status
0934:	28		:186	FLP			
0935:	00	13	:187	BNE	NO		;Desv. se nao for tempo liberado
0937:	C4	70	:188	CPY	FRETOP+1		;E o string mais baixo
0939:	00	0F	:189	BNE	NO		da memoria?
			:190	PUT	PARTE1P,01		
093B:	E4	6F	:191	CPX	FRETOP		
093D:	00	08	:192	BNE	NO		;Nao, desviar
093F:	48		:193	PHA			
0940:	18		:194	CLC			;Sim, apagar o string
0941:	65	6F	:195	ADC	FRETOP		
0943:	85	6F	:196	STA	FRETOP		
0945:	90	02	:197	BCC	BPL3		
0947:	E6	70	:198	INC	FRETOP+1		
0949:	60		:199	BPL3	PLA		
094A:	86	5E	:200	NO	STX	INDEX	;X,Y tem endereco do string
094C:	84	5F	:201	STY	INDEX+1		e A o tamanho

094E:	60	112		RTS	
		113			
094F:	C4 54	114	FRETB	CPY	LASTPT+1 ;Liberar descritor temporario
0951:	00 0C	115		ONE	RTH3
0953:	C5 53	116		CMF	LASTPT
0955:	00 0B	117		ONE	RTH3
0957:	85 52	118		STA	TEMPPT
0959:	E9 03	119		SBC	#500
095B:	85 53	120		STA	LASTPT
095D:	A0 00	121		LDY	#500
095F:	60	122	RTH3	RTS	
		123			
0960:	20 15 0A	124	CMSTR	JSR	CONCAT ;Converter a byte em X
0963:	0A	125		TAX	
0964:	40	126		PNA	;Guardar
0965:	A9 01	127		LDA	#501 ;Obter espaco para string
0967:	20 F7 06	128		JSR	STRSPA tamanho 1
096A:	68	129		PLA	;Restaurar numero
096B:	A0 00	130		LDY	#500 ;Colocar no string
096D:	91 9E	131		STA	(FAC+1),Y
096F:	68	132		PLA	
0970:	68	133		PLA	
0971:	4C 44 07	134		JMP	PUTNEW
		135			
0974:	20 03 09	136	LEFTSTR	JSR	INSTNG ;Obter parametro 1
0977:	01 0C	137		CMF	(DESCPTR),Y ;Menor que tamanho
0979:	90	138		TYA	;0(indice no comeco do string)
097A:	90 04	139	INS1	BCC	INS2 ;Desviar se parametro (tamanho
097C:	81 0C	140		LDA	(DESCPTR),Y ;Obter tamanho
097E:	AA	141		TAX	
097F:	90	142		TYA	
0980:	40	143	INS2	PNA	;Guardar indice comeco do string
0981:	0A	144	INS3	TAX	
0982:	40	145	INS4	PNA	;Guardar novo tamanho
0983:	20 F7 06	146		JSR	STRSPA ;Obter espaco para o string
0986:	A5 0C	147		LDA	DESCPTR
098B:	A4 80	148		LDY	DESCPTR+1
098A:	20 1E 09	149		JSR	FRETRP ;Liberar descritor temporario
098D:	68	150		PLA	;Obter tamanho
098E:	A0	151		TAY	
098F:	68	152		PLA	;Obter deslocamento do string
0990:	10	153		CLC	;E endereco ao pointer
0991:	65 5E	154		ADC	INDEX
0993:	85 5E	155		STA	INDEX
0995:	90 02	156		BCC	BPL4
0997:	E6 5F	157		INC	INDEX+1
0999:	98	158	BPL4	TAX	;Obter tamanho
099A:	20 00 09	159		JSR	MOVSTR ;Colocar string na area
099D:	4C 44 07	160		JMP	PUTNEW de strings
		161			
09A0:	20 03 09	162	RIGHTSTR	JSR	INSTNG ;Obter parametro 1
09A3:	10	163		CLC	
09A4:	F1 0C	164		SBC	(DESCPTR),Y ;-Tamanho -1
09A6:	49 FF	165		EDR	#FFF ;Tamanho -1
09AB:	4C 7A 09	166		JMP	INS1
		167			
09AB:	A9 FF	168	RIOSTR	LDA	#FFF ;Preparar area grande para

09A0: 85 A1	>69	STA FAC+4	eventual segundo parametro
09A1: 20 87 00	>70	JSR CHRGOT	
09A2: C9 29	>71	CHP #' '	;Tem segundo parametro?
09A4: F0 D6	>72	BEQ EMS	;Nao, desviar
09A6: 20 E6 D1	>73	JSR CHRGON	
09A9: 20 12 DA	>74	JSR GETOYT	;Obter seg. parametro em FAC+4
09BC: 20 D3 D9	>75	JSR INSTRING	;Obter primeiro parametro
09BF: CA	>76	DEX	
09C0: BA	>77	TXA	
09C1: 48	>78	PHA	;Coloc na pilha o desloc espec
09C2: 18	>79	CLC	
09C3: A2 00	>80	LDX #00	
09C5: F1 0C	>81	SBC (DISPTR),Y	;Tamanho original -1
09C7: 80 68	>82	BCC INSH	;Desv se desloc > velho tamanho
09C9: 49 FF	>83	EOB #FF	;Tamanho do resto
09CB: C5 A1	>84	CHP FAC+4	;i tamanho especificado?
09CD: 90 83	>85	BCC INSA	;Sim, desviar
09CF: A5 A1	>86	LDA FAC+4	;Obter tamanho especificado
09D1: 80 AF	>87	BCC INSA	;Sempre
	>88		
	>89		*Rotina comum a LEFTS,MIDS e RIGHTS para testar ")",
	>90		retirar da pilha o endereco de volta, obter o pointer
	>91		nde descricao e obter o primeiro parametro da instrucao
	>92		
09D3: 20 ED D1	>93	INSTRING JSR CHRGCLS	
09D6: 68	>94	PLA	;Tirar da pilha endereco
09D7: A8	>95	TAY	;de volta e guardar
09D8: 68	>96	PLA	
09D9: 85 91	>97	STA BLENGTH	
09DB: 68	>98	PLA	;Tirar da pilha endereco de
09DC: 68	>99	PLA	volta anterior (de GARGUT)
09DD: 68	>00	PLA	;Obter primeiro parametro
09DE: AA	>01	TAX	
09DF: 68	>02	PLA	
09E0: 85 0C	>03	STA DISPTR	
09E2: 68	>04	PLA	
09E3: 85 00	>05	STA DISPTR+1	
09E5: A5 91	>06	LDA BLENGTH	
09E7: 48	>07	PHA	;Por de volta na pilha
09E8: 98	>08	TYA	endereco de volta
09E9: 48	>09	PHA	
09EA: A0 00	>10	LDY #00	
09EC: 0A	>11	TXA	;Transf primeiro parametro a A
09ED: F0 10	>12	BEQ GO10	;Erro se 0
09EF: 40	>13	RTS	
	>14		
09F0: 20 F4 09	>15	LEN JSR GETSTR	
09F3: 4C 18 06	>16	JMP SGNFLT	
	>17		
09F4: 20 17 09	>18	GETSTR JSR FRESTR	
09F7: A2 00	>19	LDX #00	
09F8: 06 11	>20	STX UNLTP	
09FB: A8	>21	TAY	;Contem tamanho
09FE: 40	>22	RTS	
	>23		
09FF: 20 F6 09	>24	ASC JSR GETSTR	;Obter pointer string em INDEX
0A02: F0 00	>25	BEQ GO10	;Erro se tamanho 0

0A04:	A0 00	1126	LDY	#500		
0A06:	81 5E	1127	LDA	(INDEX),Y	;Obter primeiro caracter	
0A08:	A6	1128	TAY		;do string	
0A09:	4C 18 06	1129	JMP	SEMFLT	;Transformar em floatuante	
0A0C:	4C 83 04	1130	GO10	JMP	INERR	
		1131				
0A0F:	20 81 00	1132	STBYTIC	JSR	CHRGCT	
0A12:	20 8F 00	1133	GETBYT	JSR	FRMMUN	
0A15:	20 22 04	1134	CONINT	JSR	NRINT	
0A18:	A6 A0	1135	LDX	FAC+3	;(- 256 ?	
0A1A:	00 F0	1136	BNE	GO10	;Nao, erro	
0A1C:	A6 A1	1137	LDX	FAC+4		
0A1E:	4C 87 00	1138	JMP	CHRGCT		
		1139				
0A21:	20 F6 09	1140	VAL	JSR	GETSTR	;Obter pointer string em INDEX
0A24:	80 03	1141	BNE	VL2		
0A26:	4C 68 08	1142	JMP	ZERFAC	;Retornar 0 se tamanho=0	
0A29:	A6 88	1143	VL2	LDX	TXTPTR	
0A2B:	A4 89	1144	LDY	TXTPTR+1		
0A2D:	06 A0	1145	STX	STRING2		
0A2F:	04 AE	1146	STY	STRING2+1		
0A31:	A6 5E	1147	LDX	INDEX		
0A33:	06 88	1148	STX	TXTPTR	;Apontar TXTPTR ao comeco	
0A35:	18	1149	CLC		;do string	
0A36:	A5 5E	1150	ADC	INDEX	;Somar tamanho	
0A38:	85 A0	1151	STA	DEST	;Apontar DEST ao fim do string + 1	
0A3A:	A6 5F	1152	LDX	INDEX+1		
0A3C:	86 89	1153	STX	TXTPTR+1		
0A3E:	90 01	1154	BCC	VL3		
0A40:	E8	1155	INX			
0A41:	86 A1	1156	VL3	STX	DEST+1	
0A43:	A0 00	1157	LDY	#500		
0A45:	81 A0	1158	LDA	(DEST),Y	;Obter byte seguinte ao string	
0A47:	48	1159	PHA		;Guarda-lo	
0A48:	A9 00	1160	LDA	#500		
0A4A:	91 A0	1161	STA	(DEST),Y	;Colocar 0 no lugar	
0A4C:	20 87 00	1162	JSR	CHRGCT		
0A4F:	20 64 0F	1163	JSR	FIN	;Avaliar string	
0A52:	68	1164	PLA			
0A53:	A0 00	1165	LDY	#500		
0A55:	91 A0	1166	STA	(DEST),Y	;Trocar byte final	
0A57:	A6 A0	1167	POINT	LDX	STRING2	
0A59:	A4 AE	1168	LDY	STRING2+1		
0A5B:	86 88	1169	STX	TXTPTR		
0A5D:	84 89	1170	STY	TXTPTR+1		
0A5F:	60	1171	RTS			
		1172				
0A60:	20 BF 00	1173	GTNUM	JSR	FRMMUN	;Avaliar sintaxe: 2bytes, 1byte
0A63:	20 6C DA	1174	JSR	BETADR		;2bytes -> LNUMUN
0A66:	20 E6 01	1175	CONBYT	JSR	CHKCON	;1byte -> X
0A69:	4C 12 DA	1176	JMP	GETBYT		
		1177				
0A6C:	A5 90	1178	BETADR	LDA	FAC	;FAC != \$FFFF?
0A6E:	C9 91	1179	CMF	#591		
0A70:	80 9A	1180	BCS	GO10		;Nao, erro
0A72:	20 DC 0F	1181	JSR	QINT		;Converter a inteiro

```

DA75: A5 A0 1182 LDA FAC+3 ;E mover
DA77: A4 A1 1183 LDY FAC+4
DA79: 84 50 1184 STY LIMMUM ;A LIMMUM
DA7B: 85 51 1185 STA LIMMUM+1
DA7D: 60 1186 RTS
331 PUT PARTE14,01
31
DA7E: A5 50 12 PEEX LDA LIMMUM ;Proteger LIMMUM
DA80: 48 13 PMA
DA81: A5 51 14 LDA LIMMUM+1
DA83: 48 15 PMA
DA84: 20 6C 0A 16 JSR GETADR
DA87: A0 00 17 LDY #500
DA89: 81 50 18 LDA (LIMMUM),Y ;Fazer o PEEK
DA8B: A8 19 TAY
DA8C: 48 110 PLA ;Obter LIMMUM
DA8D: 85 51 111 STA LIMMUM+1
DA8F: 48 112 PLA
DA90: 85 50 113 STA LIMMUM
DA92: 4C 18 06 114 JMP SGMFLT ;Flutuar Y
115
DA95: 20 60 0A 116 POKE JSR GTMUM ;Obter byte para POKE em X
DA9B: 8A 117 TXA ;E endereco em LIMMUM
DA9F: A0 00 118 LDY #500
DA9B: 91 50 119 STA (LIMMUM),Y
DA9D: 60 120 RTS
121
DA9E: 20 60 0A 122 WAITB JSR GTMUM ;Obter endereco em LIMMUM
DA9F: 8A 85 123 STX FORPNT ;E mascara especifica FORPNT
DA93: A2 00 124 LDY #500
DA95: 20 87 00 125 JSR CHRGOT ;Especificado byte de inversao?
DA9B: F0 03 126 BEQ MT2 ;Nao, desviar
DA9A: 20 66 0A 127 JSR COMRYTE ;Obte-lo
DA9D: 06 86 128 MT2 STX FORPNT+1 ;Preparar byte de inversao
DA9F: A0 00 129 LDY #500
DA91: 81 50 130 MT3 LDA (LIMMUM),Y ;Obter byte no endereco
DA93: 45 86 131 EOR FORPNT+1 ;Inverter
DA95: 25 85 132 AND FORPNT ;Mascarar
DA97: F0 F0 133 BEQ MT3 ;Loop ate diferente de 0
DA99: 60 134 RTN4 RTS
135
136 *****
137 * *
138 * Rotinas de ponto *
139 * *
140 * flutuante *
141 * *
142 *****
143
DA9A: A9 7E 144 FNDGH LDA #HALF ;FAC + 1/2 -> FAC
DA9C: A0 E1 145 LDY #YHALF
DA9E: 4C 08 0A 146 JRP FAGO
147
DA91: 20 F0 0C 148 FSUB JSR COMRPE ;Carregar ARG com (A,Y)
DA94: A5 A2 149 FSUBT LDA FACSGH ;ARG - FAC -> FAC
DA96: 49 FF 150 EOR #5FF
DA98: 85 A2 151 STA FACSGH

```

DA0A: 45 AA	152		EDX	AR080H	
DA0C: 85 AB	153		STA	SGMCPH	
DA0E: A5 9B	154		LDA	FAC	
DA0B: 4C 0B DA	155		JMP	FADDT	
	156				
DA03: 20 0A DC	157	AD0	JSR	SHIFP	;Deslocar 1 byte
DA06: 90 3C	158		ICC	AI	;Sempre
DA08: 20 FB DC	159	FAD0	JSR	CONVPH	;{A,Y} em AR0
DA0E: 00 03	160	FADDT	RNE	AD1	;AR0 + FAC -> FAC
DA00: 4C 0D DE	161		JMP	NOVFA	;Se FAC= 0, so mover AR0
DAED: A6 AC	162	AD1	LTX	EXTRAFAC	;Byte a mais para precisao
DAE2: 86 92	163		STX	EXTRAGV	; em todas as rotinas FP
DAE4: A2 A5	164		LTX	MB6	;Preparar para deslocar AR0
DAE6: A5 A5	165		LDA	AR0	
DAEB: A8	166	AD2	TAY		
DAEY: FD CE	167		BEH	BTM4	;Se AR0= 0, sair
DAEB: 3B	168		SEC		
DAEC: E5 9D	169		SBC	FAC	;Obter diferenca entre expoente
DAEE: FD 24	170		BEH	AD5	;Somar se mesmo expoente
DAF0: 90 12	171		BCC	AD3	
DAF2: 84 9B	172		STY	FAC	;Troca
DAF4: 84 AA	173		L0Y	AR080H	
DAF6: 84 A2	174		STY	FAC08H	
DAF8: 4F FF	175		EDX	WFF	
DAFA: 6F 00	176		ADC	W00	
DAFC: AD 00	177		L0Y	W00	
DAFE: 84 92	178		STY	EXTRAGV	
DB00: A2 9D	179		LTX	WFA	;Preparar para deslocar FAC
DB02: D0 04	180		RNE	AD4	
DB04: AD 00	181	AD3	L0Y	W00	
DB06: 84 AC	182		STY	EXTRAFAC	
DB08: C9 F9	183	AD4	CHP	WFF	;Deslocar de quantos bits?
DB0A: 3B C7	184		BHI	AD0	;Desviar se mais que 7
DB0C: A8	185		TAY		;Indice ao num de deslocamento
DB0D: A5 AC	186		LDA	EXTRAFAC	
DB0F: 56 01	187		LSR	LOC1,X	
DB11: 20 21 DC	188		JSR	SHIFR	;Deslocar
DB14: 24 AB	189	AD5	BIT	SGMCPH	;Mesmo sinal?
DB16: 10 57	190		BPL	ADWPH	;Sim, somar as mantissas
DB18: AD 9B	191		L0Y	WFA	
DB1A: ED A5	192		CPX	MB6	;Qual esta ajustada?
DB1C: FD 02	193		BEH	DBWPH	;Se AR0, fazer FAC-AR0
DB1E: AD A5	194		L0Y	MB6	;Se FAC, fazer AR0-FAC
DB20: 3B	195	SUBWPH	SEC		
DB21: 4F FF	196		EDX	WFF	
DB23: A5 92	197		ADC	EXTRAGV	
DB25: 85 AC	198		STA	EXTRAFAC	
DB27: 8F 04 00	199		LDA	LOC4,Y	
DB2A: F5 04	1100		SBC	LOC4,X	
DB2C: 85 A1	1101		STA	FAC+4	
DB2E: 8F 03 00	1102		LDA	LOC3,Y	
DB31: F5 03	1103		SBC	LOC3,X	
DB33: 85 A0	1104		STA	FAC+3	
DB35: 8F 02 00	1105		LDA	LOC2,Y	
DB38: F5 02	1106		SBC	LOC2,X	
DB3A: 85 9F	1107		STA	FAC+2	
DB3C: 8F 01 00	1108		LDA	LOC1,Y	

0B3F:	FS 01	>109	SBC	LDC1,X	
0B41:	85 9C	>110	STA	FAC+1	
0B43:	80 03	>111	SGHIF	RCS	SGHIF ;Desviar se diferenca positiva
0B45:	20 8B 0B	>112	JBR	NEGFAC	
0B4B:	A0 00	>113	SGHIF	LDF	#500 ;Deslocar digito significativo
0B4A:	98	>114	TYA		;contando deslocamento em A
0B4B:	18	>115	CLC		
0B4C:	A6 9C	>116	LOOP	LDF	FAC+1
0B4E:	D0 46	>117	BNE	FR2	;Repetir ate diferente de 0
0B50:	A6 9F	>118	LDF	FAC+2	
0B52:	B6 9E	>119	STX	FAC+1	
0B54:	A6 A0	>120	LDF	FAC+3	
0B5A:	B6 9F	>121	STX	FAC+2	
0B58:	A6 A1	>122	LDF	FAC+4	
0B5A:	B6 A0	>123	STX	FAC+3	
0B5C:	A6 AC	>124	LDF	EXTRAFAC	
0B5E:	B6 A1	>125	STX	FAC+4	
0B60:	B4 AC	>126	STY	EXTRAFAC	;Zerar byte a mais
0B62:	A9 0B	>127	ADC	#500	;Contar os 8 bits
0B64:	C9 20	>128	CMF	#B04	;Feito 4 vezes?
0B66:	00 E4	>129	BNE	LOOP	;Mas, loop
0B68:	A9 00	>130	ZEROFAC	LDA	#500
0B6A:	B5 9D	>131	ALO/FAC	STA	FAC
0B6C:	B5 A2	>132	ALO/FACS	STA	FACSGN
0B6E:	A0	>133		RTS	
		>134			
0B6F:	A5 92	>135	ADDAH	ADC	EXTRASV ;Somar mantissas
0B71:	B5 AC	>136	STA	EXTRAFAC	
0B73:	A5 A1	>137	LDA	FAC+4	
0B75:	A5 A9	>138	ADC	ARG+4	
0B77:	B5 A1	>139	STA	FAC+4	
0B79:	A5 A0	>140	LDA	FAC+3	
0B7B:	A5 A8	>141	ADC	ARG+3	
0B7D:	B5 A0	>142	STA	FAC+3	
0B7F:	A5 9F	>143	LDA	FAC+2	
0B81:	A5 A7	>144	ADC	ARG+2	
0B83:	B5 9F	>145	STA	FAC+2	
0B85:	A5 9C	>146	LDA	FAC+1	
0B87:	A5 A6	>147	ADC	ARG+1	
0B89:	B5 9C	>148	STA	FAC+1	
0B8B:	4C A7 0B	>149	JMP	FR3	
		>150			
0B8E:	A9 01	>151	FR1	ADC	#501 ;Contar bits deslocados
0B90:	06 AC	>152	ASL	EXTRAFAC	
0B92:	26 A1	>153	RDL	FAC+4	
0B94:	26 A0	>154	RDL	FAC+3	
0B96:	26 9F	>155	RDL	FAC+2	
0B98:	26 9C	>156	RDL	FAC+1	
0B9A:	10 F2	>157	FR2	BPL	FR1 ;Repetir ate FAC+1 negativo
0B9C:	3B	>158	SEC		
0B9E:	E3 9D	>159	SBC	FAC	;Fixar expoente
0B9F:	80 C7	>160	RCS	ZEROFAC	
0BA1:	49 FF	>161	EDM	#5FF	
0BA3:	A9 01	>162	ADC	#501	
0BA5:	B5 9D	>163	STA	FAC	;Carry e desligado aqui
0BA7:	9D 0E	>164	FR3	RCC	BTMS
0BA9:	E4 9D	>165	PRODM	JMC	FAC

DBA0:	F0 42	1166	BEQ	OVERFLOW	
DBA0:	66 9E	1167	ROR	FAC+1	
DBA0:	66 9F	1168	ROR	FAC+2	
DBA1:	66 A0	1169	ROR	FAC+3	
DBA3:	66 A1	1170	ROR	FAC+4	
DBA5:	66 AC	1171	ROR	EXTRAFAC	
DBE7:	60	1172	RTNS	RTS	
		1173			
DBB9:	A5 A2	1174	NEGFA	LDA	FACSDN ;Fazer complemento a um
DBBA:	49 FF	1175	EDR	MSFF	
DBBC:	85 A2	1176	STA	FACSDN	
DBBE:	A5 9E	1177	NEG2	LDA	FAC+1
DBC0:	49 FF	1178	EDR	MSFF	
DBC2:	85 9E	1179	STA	FAC+1	
DBC4:	A5 9F	1180	LDA	FAC+2	
DBC6:	49 FF	1181	EDR	MSFF	
DBC8:	85 9F	1182	STA	FAC+2	
DBCA:	A5 A0	1183	LDA	FAC+3	
DBCC:	49 FF	1184	EDR	MSFF	
DBCE:	85 A0	1185	STA	FAC+3	
DBD0:	A5 A1	1186	LDA	FAC+4	
DBD2:	49 FF	1187	EDR	MSFF	
DBD4:	85 A1	1188	STA	FAC+4	
DBD6:	A5 AC	1189	LDA	EXTRAFAC	
DBD8:	49 FF	1190	EDR	MSFF	
DBDA:	85 AC	1191	STA	EXTRAFAC	
DBDC:	E6 AC	1192	INC	EXTRAFAC	;Somar bit para obter o
DBDE:	00 0E	1193	SME	RTN6	complemento a 2
DBE0:	E6 A1	1194	PLUS8PS	INC	FAC+4 ;Somar carry de EXTRA
DBE2:	00 0A	1195	BNE	RTN6	
DBE4:	E6 A0	1196	INC	FAC+3	
DBE6:	00 0A	1197	SME	RTN6	
DBE8:	E6 9F	1198	INC	FAC+2	
DBEA:	00 02	1199	BNE	RTN6	
DBEC:	E6 9E	1200	INC	FAC+1	
DBEE:	60	1201	RTN6	RTS	
		1202			
DBF1:	A2 40	1203	OVERFLOW	LDB	NESTOR-HEMSEIRO
DBF1:	4C 01 C7	1204	JMP	ERROR	
		302	PUT	PARTEIX,01	
		31			
DBF4:	A2 61	32	SHIFTR	LDX	MSRESULT-1 ;Entrada de FNULT
DBF6:	X4 04	33	KISFT	LDY	LOC4,X ;Carry deve ter sido ligada
DBF8:	84 AC	34	STY	EXTRAFAC	
DBFA:	84 03	35	LDY	LOC3,X	
DBFC:	94 04	36	STY	LOC4,X	
DBFE:	84 02	37	LDY	LOC2,X	
DC00:	94 03	38	STY	LOC3,X	
DC02:	X4 01	39	LDY	LOC1,X	
DC04:	94 02	40	STY	LOC2,X	
DC06:	A4 A4	41	LDY	FPBEN	;IFF se vem de 0INT p/ num. neg
DC08:	94 01	42	STY	LOC1,X	;Caso contrario 0
DC0A:	49 08	43	SHOFT	ADC	MSOB ;Desloc 1,X a dir \$100-4 bits
DC0C:	30 E8	44	BNE	KISFT	;fazer deslocamento de byte
DC0E:	F0 E6	45	BEQ	KISFT	se esta no intervalo
DC10:	E9 08	46	SBC	MSOB	
DC12:	A0	47	TAY		;Contar para deslocamento

DC13: 85 AC	>18	LDA	EXTRAFAC	Final de bit
DC15: 80 14	>19	BOS	SH3	;Sair se não precisa de desloc
DC17: 16 01	>20	SH1	ASL LDC1,X	;Deslocar somente os 7 bits
DC19: 90 02	>21	BCC	SH2	de baixa ordem de 1,X
DC1B: F4 01	>22	INC	LDC1,X	;Forçar próxima instrução
DC1D: 74 01	>23	SH2	ROR LDC1,X	a ligar carry
DC1F: 74 01	>24	ROR	LDC1,X	
DC21: 74 02	>25	SHFTL	ROR LDC2,X	
DC23: 74 03	>26	ROR	LDC3,X	
DC25: 74 04	>27	ROR	LDC4,X	
DC27: 6A	>28	ROR		
DC2B: C3	>29	INP		
DC2F: 00 EC	>30	RNE	SH1	
DC2B: 18	>31	SH3	CLC	
DC2C: 60	>32		RTS	
	>33			
DC2D: 81 00 00	>34	HLM	HEX	8100000000
DC30: 00 00				
DC32: 03	>35	LOGSER	HEX	03 ;Índice do número de coeficientes
DC33: 7F 5E 56	>36		HEX	7F5E560B7F
DC36: C8 79				
DC3B: 80 13 96	>37		HEX	80139603e4
DC3B: 08 e4				
DC3D: 80 74 38	>38		HEX	8074387316
DC40: 93 16				
DC42: 62 38 4A	>39		HEX	62384A3B2D
DC45: 38 20				
DC47: 80 35 04	>40	SERdecio	HEX	803504F334 ;SAR(1,1/2)
DC4A: F3 34				
DC4C: 81 35 04	>41	SERdecis	HEX	813504F334 ;SAR(1/2)
DC4F: F3 34				
DC51: 60 60 00	>42	Resumo	HEX	0060000000 ;1-1/2
DC54: 00 00				
DC56: 80 35 72	>43	LOGdecis	HEX	80357217FB ;LN(2)
DC59: 17 F8				
	>44			
DC5B: 20 9C 0E	>45	LOG	JSR	SIGN ;Logaritmo natural de FAC
DC5E: F0 02	>46	BER	G18	;Argumento deve ser > 0
DC60: 10 03	>47		RPL	L02
DC62: 4C 83 04	>48	G18	JMP	[ERR
DC65: A5 9D	>49	L02	LDA	FAC ;Guardar expoente = 680
DC67: E9 7F	>50		SBC	#57F ;Limpar carry
DC6F: 48	>51		PhA	
DC6A: A9 8D	>52		LDA	#68D ;Normalizar entre .5 e 1
DC6C: 85 9D	>53		STA	FAC
DC6E: A9 47	>54		LDA	#58Resumo
DC70: A0 0C	>55		LDY	#58Resumo
DC72: 20 08 0A	>56		JSR	FAC0 ;Calc na serie de pot. , impares
DC75: A9 4C	>57		LDA	#58Resis de (58(12)(1-1)/(58(12)(1+1)
DC77: A0 0C	>58		LDY	#58Resis
DC79: 20 80 00	>59		JSR	FDIV
DC7C: A9 20	>60		LDA	#HUM
DC7E: A0 0C	>61		LDY	#HUM
DC80: 20 C1 06	>62		JSR	FSUM
DC83: A9 32	>63		LDA	#LOGSER
DC85: A0 0C	>64		LDY	#LOGSER
DC87: 20 76 E2	>65		JSR	000SER ;Calcular LOG +.5 base 2

DC8A: A9 51	166	LDA	MENACIO	
DC8C: A0 0C	167	LDY	R/MENACIO	
DC8E: 20 08 0A	168	JSR	FADD	
DC91: 68	169	PLA		
DC92: 20 EF DF	170	JSR	ADDA0C	;Somar expoente original - 500
DC95: A9 5A	171	LDA	MLOG60is	;Obtido LOG base 2, converter a
DC97: A0 0C	172	LDY	R/MLOG60is	base e multiplic. por LOG(2)
DC99: 20 FD 0C	173	FNULT	JSR	COMUPE ;(A,Y) em ARG
DC9C: 00 03	174	FNULTT	BNE	FNU ;FAC * ARG -> FAC
DC9E: AC FC 0C	175	JMP	RTM7	
	176			
DCA1: 20 28 00	177	FNU	JSR	ADEXP
DCA4: A9 00	178	LDA	MS00	;Inicializar produto
DCA6: 85 62	179	STA	RESULT	
DCA8: 85 63	180	STA	RESULT+1	
DCAA: 85 64	181	STA	RESULT+2	
DCAC: 85 65	182	STA	RESULT+3	
DCAE: A5 AC	183	LDA	EXTRAFAC	;Multiplicar digitos de FAC por
DCB0: 20 CA 0C	184	JSR	FH1	ARG e somar a RESULT
DCB3: A5 A1	185	LDA	FAC+4	
DCB5: 20 CA 0C	186	JSR	FH1	
DCB8: A5 A0	187	LDA	FAC+3	
DCBA: 20 CA 0C	188	JSR	FH1	
DCBD: A5 9F	189	LDA	FAC+2	
DCBF: 20 CA 0C	190	JSR	FH1	
DCC2: A5 9E	191	LDA	FAC+1	
DCC4: 20 CF 0C	192	JSR	FM2	
DCC7: AC 00 0E	193	JMP	RES/FAC	;Mov RESULT,a FAC e normalizar
	194			
	195			Rotina para multiplicar A por ARG e somar a RESULT
	196			
DCCA: 00 03	197	FH1	BNE	FM2 ;Fazer mult.de 8 bits se nao 0
	198			
DCCC: AC F4 0E	199	JMP	SHFTRES	;Desloc.o prod.de um bjt.p/ace1.
DCCF: A4	1100	FM2	LSR	;Desloc.p./fora b t de baixa ord.
DCD0: 09 80	1101		ORA	MS00 ;Prez para contar 8 bits
DCD2: A8	1102	FM3	TAX	;Guardar
DCD3: 90 19	1103		BCC	FMA ;Desviar se bit de baixa ord 0
DCD5: 18	1104		CLC	;Multar 1 bit por ARG em RESULT
DCD6: A5 65	1105	LDA	RESULT+3	
DCD8: A5 66	1106	ADC	ARG+4	
DCDA: 85 65	1107	STA	RESULT+3	
DCDC: A5 64	1108	LDA	RESULT+2	
DCDE: A5 68	1109	ADC	ARG+3	
DCDF: 85 64	1110	STA	RESULT+2	
DCE2: A5 63	1111	LDA	RESULT+1	
DCE4: A5 67	1112	ADC	ARG+2	
DCE6: 85 63	1113	STA	RESULT+1	
DCE8: A5 62	1114	LDA	RESULT	
DCEA: A5 A6	1115	ADC	ARG+1	
DCEC: 85 62	1116	STA	RESULT	
DCEE: 66 62	1117	FM4	ROR	RESULT ;Deslocar o produto de 1 bit
DCF0: 66 63	1118	ROR	RESULT+1	
DCF2: 66 64	1119	ROR	RESULT+2	
DCF4: 66 65	1120	ROR	RESULT+3	
DCF6: 66 AC	1121	ROR	EXTRAFAC	
DCF8: 98	1122	TAX		;Obter acumulado

0CF9:	4A	:123	LSR		Deslocar para fora pros bit
0CFA:	00 56	:124	RNE	FAC	Loop 8 vezes (via ORA #80)
0CFC:	60	:125	RTN?	RTS	
		:126			
		:127			Descompactar numero em (A,Y) e mover a ARG
		:128			
0CFD:	85 5E	:129	CMULPX	STA INDEX	
0CFF:	84 5F	:130	STY	INDEX+1	
0D01:	A0 04	:131	LDT	#504	
0D03:	81 5E	:132	LDA	(INDEX),Y	
0D05:	85 A9	:133	STA	ARG+4	
0D07:	88	:134	DEY		
0D08:	81 5E	:135	LDA	(INDEX),Y	
0D0A:	85 A8	:136	STA	ARG+3	
0D0C:	88	:137	DEY		
0D0D:	81 5E	:138	LDA	(INDEX),Y	
0D0F:	85 A7	:139	STA	ARG+2	
0D11:	88	:140	DEY		
0D12:	81 5E	:141	LDA	(INDEX),Y	
0D14:	85 AA	:142	STA	ARGSGN	Guardar sinal
0D16:	4C A2	:143	EOR	FACSGN	
0D18:	85 A8	:144	STA	SGNCPR	Preparar comparacao de sinais
0D1A:	85 AA	:145	LDA	ARGSGN	Obter bit mais significativo
0D1C:	09 80	:146	ORA	#80	Preparar primeiro bit
0D1E:	85 A6	:147	STA	ARG+1	Guardar bit mais significativo
0D20:	88	:148	DEY		
0D21:	81 5E	:149	LDA	(INDEX),Y	
0D23:	85 A5	:150	STA	ARG	Guardar expoente
0D25:	45 90	:151	LDA	FAC	para registro de status
0D27:	60	:152	RTS		
		:153			
0D28:	45 A5	:154	ADEXP	LDA ARG	
0D2A:	F0 1F	:155	ADEX12	BCR ZERO	
0D2C:	10	:156	CLC		
0D2D:	45 90	:157	ADC	FAC	
0D2F:	90 04	:158	RCC	ADEX3	Desviar se nao houver overflow
0D31:	30 10	:159	BHI	JOV	
0D33:	18	:160	CLC		OK, +530 nao vai dar overflow
0D34:	2C	:161	HEX	2C	
0D35:	10 14	:162	ADEX3	BPL ZERO	Se ainda positivo, underflow
0D37:	6F 80	:163	ADC	#580	Corrigir para deslocamento 580
0D39:	85 90	:164	STA	FAC	
0D3B:	00 03	:165	RNE	ADEX4	
0D3D:	4C 6C 08	:166	JMP	#0FAC5	
0D3F:	45 A8	:167	ADEX4	LDA SGNCPR	
0D41:	85 A2	:168	STA	FACSGN	
0D43:	60	:169	RTS		
		:170			
0D45:	45 A2	:171	OUTOFBNS	LDA FACSGN	
0D47:	4F FF	:172	EOR	#5FF	
0D49:	30 05	:173	BHI	JOV	Erro se numero positivo
0D4B:	68	:174	ZERO	PLA	
0D4D:	68	:175		PLA	
0D4F:	4C 68 08	:176	JMP	ZEROFAC	Retornar 0 se numero negativo
0D51:	4C EF 08	:177	JOV	JMP OVERFLOW	
		:178			
		:179			*Rotina para multiplicar FAC por 10

```

>180
0053: 20 7D DE >181 MULD JSR MOVWF ;Copiar FAC em ARG
0056: AA >182 TAN ;A contem FAC
0057: FD 10 >183 BEQ RTMB ;Sair se FAC = 0
0059: 18 >184 CLC
005A: 69 02 >185 ADC #02 ;Simular #4
005C: 80 F2 >186 BCS JNV
005E: A2 00 >187 LDX #00 ;Indica soma de operandos
0060: 86 A8 >188 STX SGNCPR de mesmo sinal
0062: 20 E0 0A >189 JSR AD2 ;FAC A4 + ARG -> ARG
0065: E6 90 >190 INC FAC ;= multiplicacao por 2
0067: FD E7 >191 BEQ JNV
0069: 60 >192 RTMB RTS
>193
006A: 84 20 D0 >194 MULD HEX B420000000
006D: D0 D0 >195
>196
>196 ;Rotina para dividir ARG(FAC) por 10
>197
006F: 20 7D DE >198 DIVD JSR MOVWF ;Copiar FAC em ARG
0072: A9 6A >199 LDA #0000 ;Preparar para por 10 em FAC
0074: A0 10 >200 LDY #10000
0076: A2 00 >201 LDX #00
0078: 86 A8 >202 STX SGNCPR
007A: 20 13 DE >203 JSR MOVWF ;Colocar (A,Y) em FAC
007D: 4C E3 00 >204 JMP FDIY1 ;Dividir ARG por FAC
>205
>205 PUT PARTE15,D1
>206
0080: 20 FD DC >207 FDIY JSR CONVPK ;(A,Y) -> ARG
0083: FD 7A >208 FDIY BEQ DIV2 ;ARG/FAC -> FAC
0085: 20 8C DE >209 JSR RMDB
0088: A9 00 >210 LDA #000
008A: 38 >211 SEC
008B: E5 90 >212 SBC FAC
008D: B5 90 >213 STA FAC
008F: 20 28 D0 >214 JSR ADEXP ;Obter expoente de ARG/(2*FAC)
0092: E6 90 >215 INC FAC ;#2
0094: FD 8A >216 BEQ JNV
0096: A2 FC >217 LDX #4 ;Indice do loop
0098: A9 01 >218 LDA #01 ;Contador de bits e quoc. parc.
009A: 84 A6 >219 LDY ARG+1 ;ARG := FAC*
009C: C4 9E >220 CPY FAC+1
009E: D0 10 >221 BNE FD2
00A0: A4 A7 >222 LDY ARG+2
00A2: C4 9F >223 CPY FAC+2
00A4: D0 0A >224 BNE FD2
00A6: A4 A8 >225 LDY ARG+3
00A8: C4 A0 >226 CPY FAC+3
00AA: D0 0A >227 BNE FD2
00AC: A4 A9 >228 LDY ARG+4
00AE: C4 A1 >229 CPY FAC+4
00B0: 08 >230 PHP ;Sim, ligar carry
00B1: 2A >231 ROL ;Subir contador de bits e rodar
>232 ;bit quociente
00B2: 90 D9 >233 BCC FD3 ;Saltar ate completar os 8 bits
00B4: EB >234 INX ;Subir indice do loop
00B5: 95 65 >235 STA RESULT+3,X ;Guardar um byte do quociente

```

0007:	FD 32	131	BEH	FD6	;Desviar se ultimo
0009:	10 34	132	BPL	FD7	;Saida final quando Z=1
000B:	A9 01	133	LDA	#001	;Desligar contador de bits
000D:	28	134	PLP		;ARG >= FAC?
000E:	80 0E	135	BDS	FD5	;Subtrai divisor se for o caso
000D:	06 A9	136	FD4	ASL	ARG+4
0002:	26 A8	137	BRL	ARG+3	;Deslocar ARG de um bit
00C4:	26 A7	138	BRL	ARG+2	
00C6:	26 A6	139	BRL	ARG+1	
09C8:	80 E6	140	BDS	FD2	;Desv se novo ARG da overflow
00CA:	30 CE	141	BRI	FD1	;Testar se e possivel a divisao
00CC:	10 E2	142	BPL	FD2	;Mas e necessario comparacao
00CE:	A8	143	FD5	TAY	;Proteger quociente parcial
00CF:	A5 A9	144	LDA	ARG+4	
00D1:	E5 A1	145	SBC	FAC+4	
00D3:	85 A9	146	STA	ARG+4	
00D5:	A5 A8	147	LDA	ARG+3	
00D7:	E5 A0	148	SBC	FAC+3	
00D9:	85 A8	149	STA	ARG+3	
00DB:	A5 A7	150	LDA	ARG+2	
00DD:	E5 9F	151	SBC	FAC+2	
00DF:	85 A7	152	STA	ARG+2	
00E1:	A5 A6	153	LDA	ARG+1	
00E3:	E5 9E	154	SBC	FAC+1	
00E5:	85 A6	155	STA	ARG+1	
00E7:	98	156	TYA		
00E9:	4C C0 D0	157	JMP	FD4	
00EB:	A9 40	158	FD6	LDA	#540
00ED:	80 CE	159	BNE	FD3	;Prep cont de bits para ultimo
		160			;Seapre
00EF:	0A	161	FD7	ASL	
00F0:	0A	162	ASL		
00F1:	0A	163	ASL		
00F2:	0A	164	ASL		
00F3:	0A	165	ASL		
00F4:	0A	166	ASL		
00F5:	85 AC	167	STA	EXTRAFAC	;Os ult 2 bits vao a EXTRAFAC
00F7:	28	168	PLP		
00F9:	4C D0 DE	169	JMP	RESIFAC	
00FB:	A2 B6	170	OTVZ	LDX	#014po2RD-MEMSENDO
00FD:	4C D1 C7	171	JMP	ERROR	
00DF:	A5 62	172	RESIFAC	LDA	RESULT
0ED2:	85 9E	173	STA	FAC+1	
0ED4:	A5 63	174	LDA	RESULT+1	
0ED6:	85 9F	175	STA	FAC+2	
0ED8:	A5 64	176	LDA	RESULT+2	
0EDA:	85 A0	177	STA	FAC+3	
0EDC:	A5 65	178	LDA	RESULT+3	
0EDE:	85 A1	179	STA	FAC+4	
0EDF:	4C DE D8	180	JMP	SIGNDF	
		181			
		182			;Rotina para obter numero ponto flutuante compactado
		183			new GA,YI, descompactar e mover a FAC
		184			
DE13:	85 SE	185	MOVFN	STA	INDEX
DE15:	84 5F	186	STY	INDEX+1	
DE17:	A0 84	187	LDD	#504	

DE19: 81 5E	188	LDA	(INDEX),Y		
DE1B: 85 A1	189	STA	FAC+4		
DE1D: 88	190	DEY			
DE1E: 81 5E	191	LDA	(INDEX),Y		
DE20: 85 A0	192	STA	FAC+3		
DE22: 88	193	DEY			
DE23: 81 5E	194	LDA	(INDEX),Y		
DE25: 85 9F	195	STA	FAC+2		
DE27: 88	196	DEY			
DE2B: 81 5E	197	LDA	(INDEX),Y		
DE2A: 85 A2	198	STA	FACSGH	;Descompactar	
DE2C: 09 80	199	ORA	#80		
DE2E: 85 9E	100	STA	FAC+1		
DE30: 88	101	DEY			
DE31: 81 5E	102	LDA	(INDEX),Y		
DE33: 85 90	103	STA	FAC		
DE35: 84 AC	104	STY	EXTRAFAC	;Y= 0	
DE37: 60	105	RTS		;Status de acordo com FAC	
	106				
DE3B: A2 9B	107	MOV2F	LOX	ITEMP2	;Compactar FAC em TEMP2
DE3A: 2C	108	HEX	2C		
DE3F: A2 93	109	MOV1F	LOX	ITEMP1	;Compactar FAC em TEMP1
DE3D: A0 00	110	MOVNL	LDY	#00	;Byte de alta ordem do endereço
DE3F: F0 04	111	BEQ	MOVNF		de destino = 0
DE41: A6 85	112	SETFOR	LOX	FORPNT	;Chamado por LET e NEXT
DE43: A4 86	113		LDY	FORPNT+1	
DE45: 20 0C DE	114	MOVNF	JSR	END0	;Compactar FAC na memória (X,Y)
DE4B: 84 5E	115		STX	INDEX	
DE4A: 84 5F	116		STY	INDEX+1	
DE4C: A0 04	117		LDY	#04	
DE4E: A5 A1	118		LDA	FAC+4	
DE50: 91 5E	119		STA	(INDEX),Y	
DE52: 88	120	DEY			
DE53: A5 A0	121		LDA	FAC+3	
DE55: 91 5E	122		STA	(INDEX),Y	
DE57: 88	123	DEY			
DE5B: A5 9F	124		LDA	FAC+2	
DE5A: 91 5E	125		STA	(INDEX),Y	
DE5C: 88	126	DEY			
DE5D: A5 A2	127		LDA	FACSGH	
DE5F: 09 7F	128		ORA	#7F	
DE61: 25 9E	129	AND	FAC+1		
DE63: 91 5E	130		STA	(INDEX),Y	
DE65: 88	131	DEY			
DE6A: A5 90	132		LDA	FAC	
DE6B: 91 5E	133		STA	(INDEX),Y	
DE6A: 84 AC	134	STY	EXTRAFAC	;Y=0	
DE6C: 60	135	BTS			
	136				
DE6D: A5 A4	137	MOVFA	LDA	ARGSGH	;Mover ARG a FAC
DE6F: 85 A2	138	NFA	STA	FACSGH	;Entrada de FPMRT para
DE71: A2 05	139		LOX	#05	fazer ARG antes
DE73: 85 A4	140	NFA2	LDA	ARG-1,X	
DE75: 95 9C	141		STA	FAC-1,X	
DE77: 0A	142		DEX		
DE7B: 00 F9	143	BNE	NFA2		
DE7A: 86 AC	144		STX	EXTRAFAC	

DE7C:	40	1145		RTS	
		1146			
DE7D:	20 BC DE	1147	NOVNF	JSR RND6	;Arredondar, depois
DE8D:	A2 DA	1148	NAF	LDR #504	; mover FAC a ARG
DE82:	85 9C	1149	NAF2	LDA FAC+1,X	; inclusive sinal
DE84:	95 A4	1150		STA ARG+1,X	
DE86:	CA	1151		DEX	
DE87:	D0 F9	1152		RNE NAF2	
DE89:	B6 AC	1153		STX EXTRNFAC	
DE8B:	40	1154	RTW	RTS	
		1155			
		1156	;Rotina de uso geral para arredondar FAC usando os		
		1157	;bits mais significativos de EXTRNFAC		
		1158			
DE8C:	45 90	1159	RND6	LDA FAC	;Evitar se numero=0
DE8E:	F0 F8	1160	REB	RTW	
DE9D:	D6 AC	1161	ASL	EXTRNFAC	;se EXTRNFAC negativo
DE92:	90 F7	1162	BCC	RTW	;Somar, um bit
DE94:	20 E0 08	1163	ROUND	JSR PLUS6PS	;o numero em FAC
DE97:	00 F2	1164	RNE	RTW	
DE99:	4C A9 D8	1165	JMP	FROUND	;Arredondar se expoente afetado
		1166			
DE9C:	A5 90	1167	SIGN	LDA FAC	;testar sinal de FAC e
DE9E:	F0 09	1168	REB	RTW	retornar -1,0,1 em A
DEA0:	45 A2	1169	SIGN1	LDA FACSIGN	de acordo com o resultado
DEA2:	2A	1170	SIGN2	RDL	
DEA3:	A9 FF	1171		LDA #5FF	
DEA5:	80 02	1172	RCS	RND10	
DEA7:	A9 D1	1173		LDA #5D1	
DEA9:	40	1174	RTNDD	RTS	
		1175			
DEAA:	20 9C DE	1176	SGN	JSR SIGN	;Converter FAC a -1,0,1
DEAD:	85 9E	1177	FLDNT	STA FAC+1	;Converter a flutuante
DEAF:	A9 00	1178		LDA #5D0	conteudo de A
DEB1:	85 9F	1179		STA FAC+2	
DEB3:	A2 88	1180		LDR #588	;OR a direita 8 bits
DEB5:	A5 9E	1181	FLD1	LDA FAC+1	;Entrada de DINTF para
DEB7:	A9 FF	1182		EOR #5FF	conv.a flut.de 2 bytes c/sig.
DEB9:	2A	1183	RDL		;Ligar carry se numero positivo
DEBA:	A9 00	1184	FLD2	LDA #5D0	;Entr de LIMPRT para converter
DEBC:	85 A1	1185		STA FAC+4	a flutuante int de 2 bytes
DEBE:	95 A0	1186		STA FAC+3	sem sinal
DECD:	B6 90	1187		STX FAC	;Preparar expoente
DECE:	85 AC	1188		STA EXTRNFAC	;limpar byte extra
DECA:	85 A2	1189		STA FACSIGN	;Faz-lo positivo
DECC:	4C A3 D8	1190	JMP	SGNIF	;ajust. sinal e bit mais signif
		1191			
DECP:	46 A2	1192	ABS	LSR FACSIGN	;mudar sinal para positivo
DECB:	40	1193		RTS	
		1194			
		1195			
		1196			
		1197			
		1198			
		1199			
		1200			
		1201			
		1202			
		1203			
		1204			
		1205			
		1206			
		1207			
		1208			
		1209			
		1210			
		1211			
		1212			
		1213			
		1214			
		1215			
		1216			
		1217			
		1218			
		1219			
		1220			
		1221			
		1222			
		1223			
		1224			
		1225			
		1226			
		1227			
		1228			
		1229			
		1230			
		1231			
		1232			
		1233			
		1234			
		1235			
		1236			
		1237			
		1238			
		1239			
		1240			
		1241			
		1242			
		1243			
		1244			
		1245			
		1246			
		1247			
		1248			
		1249			
		1250			
		1251			
		1252			
		1253			
		1254			
		1255			
		1256			
		1257			
		1258			
		1259			
		1260			
		1261			
		1262			
		1263			
		1264			
		1265			
		1266			
		1267			
		1268			
		1269			
		1270			
		1271			
		1272			
		1273			
		1274			
		1275			
		1276			
		1277			
		1278			
		1279			
		1280			
		1281			
		1282			
		1283			
		1284			
		1285			
		1286			
		1287			
		1288			
		1289			
		1290			
		1291			
		1292			
		1293			
		1294			
		1295			
		1296			
		1297			
		1298			
		1299			
		1300			
		1301			
		1302			
		1303			
		1304			
		1305			
		1306			
		1307			
		1308			
		1309			
		1310			
		1311			
		1312			
		1313			
		1314			
		1315			
		1316			
		1317			
		1318			
		1319			
		1320			
		1321			
		1322			
		1323			
		1324			
		1325			
		1326			
		1327			
		1328			
		1329			
		1330			
		1331			
		1332			
		1333			
		1334			
		1335			
		1336			
		1337			
		1338			
		1339			
		1340			
		1341			
		1342			
		1343			
		1344			
		1345			
		1346			
		1347			
		1348			
		1349			
		1350			
		1351			
		1352			
		1353			
		1354			
		1355			
		1356			
		1357			
		1358			
		1359			
		1360			
		1361			
		1362			
		1363			
		1364			
		1365			
		1366			
		1367			
		1368			
		1369			
		1370			
		1371			
		1372			
		1373			
		1374			
		1375			
		1376			
		1377			
		1378			
		1379			
		1380			
		1381			
		1382			
		1383			
		1384			
		1385			
		1386			
		1387			
		1388			
		1389			
		1390			
		1391			
		1392			
		1393			
		1394			
		1395			
		1396			
		1397			
		1398			
		1399			
		1400			
		1401			
		1402			
		1403			
		1404			
		1405			
		1406			
		1407			
		1408			
		1409			
		1410			
		1411			
		1412			
		1413			
		1414			
		1415			
		1416			
		1417			
		1418			
		1419			
		1420			
		1421			
		1422			
		1423			
		1424			
		1425			
		1426			
		1427			
		1428			
		1429			
		1430			
		1431			
		1432			
		1433			
		1434			
		1435			
		1436			
		1437			
		1438			
		1439			
		1440			
		1441			
		1442			
		1443			
		1444			
		1445			
		1446			
		1447			
		1448			
		1449			
		1450			
		1451			
		1452			
		1453			
		1454			
		1455			
		1456			
		1457			
		1458			
		1459			
		1460			
		1461			
		1462			
		1463			
		1464			
		1465			
		1466			
		1467			
		1468			
		1469			
		1470			
		1471			
		1472			

DED4: C8	18		INY	
DED5: A6	19		TAX	
DED6: FD C4	110		BEG SIGH	;Desviar se (A,Y) for zero
DED8: B1 60	111		LDA (DEST),Y	
DED9: 45 A2	112		ERR FACSGH	
DEDC: 30 C2	113		RMI SIGH1	;Desviar se sinais diferentes
DEDE: E4 90	114		CFX FAC	
DEE0: 00 21	115		BNE FC1	;Desviar se expoentes diferentes
DEE2: B1 60	116		LDA (DEST),Y	;Descompactar e comparar
DEE4: 09 80	117		ORA #80	
DEE6: C5 9E	118		CMP FAC+1	
DEE8: 00 19	119		BNE FC1	
DEEA: C3	120		INY	
DEEB: B1 60	121		LDA (DEST),Y	
DEED: C5 9F	122		CMP FAC+2	
DEEF: 00 12	123		BNE FC1	
DEF1: CB	124		INY	
DEF2: B1 60	125		LDA (DEST),Y	
DEF4: C5 A0	126		CMP FAC+3	
DEF6: 00 00	127		BNE FC1	
DEF8: CB	128		INY	
DEF9: A9 7F	129		LDA #57F	;Usar bit extra de FAC para
DEFB: C5 AC	130		CMP EXTRAFAC	determinar carry para a
DEFD: B1 60	131		LDA (DEST),Y	ultima comparacao
DEFF: E5 A1	132		SBC FAC+4	
DF01: FD 28	133		BEB RTN11	;Saida se numeros iguais
DFD3: A5 A2	134	FC1	LDA FACSGH	
DFD5: 90 02	135		BCC FC2	;Desviar se (A,Y) < FAC em
DFD7: 49 FF	136		ERR #FF	valor absoluto
DFD9: 4C A2 0E	137	FC2	JMP SIGH2	
	138			
	139			##Na saída de FCMP, A=1,0,-1 conforme (A,Y) (<,>,>) FAC
	140			
DFDC: A5 90	141	RTN1	LDA FAC	;Converter FAC a sua parte int
DFDE: FD 46	142		BEG ZFAC	
DF10: 38	143		SEC	;Supor FAC > 2^31
DF11: E9 A0	144		SBC #5A0	;Resultado em
DF13: 24 A2	145		BIT FACSGH	FAC+1 a FAC+4
DF15: 10 09	146		RPL #11	;((9E-5A2)
DF17: A6	147		TAX	
DF18: A9 FF	148		LDA #FF	
DF1A: B5 A1	149		STA FPGEN	
DF1C: 20 BE 06	150		JSR MED2	
DF1F: 8A	151		TXA	
DF20: A2 90	152	#11	LDX #FAC	
DF22: C9 F9	153		CMP #5F9	;Mais de 7 bits a deslocar?
DF24: 10 06	154		RPL #12	;Não, desviar
DF26: 20 0A 0C	155		JSR SHIFT	;Sim, realizar desloc do byte
DF29: B4 A4	156		STY FPGEN	;Y=0
DF2B: 60	157	RTN11	RTS	
	158			
DF2C: A8	159	#12	TAY	;Numero de bits a deslocar
DF2D: A5 A2	160		LDA FACSGH	
DF2F: 29 80	161		AND #80	;08ter sinal
DF31: 46 9E	162		LSR FAC+1	
DF33: C5 9E	163		ORA FAC+1	
DF35: B5 9E	164		STA FAC+1	;Restabelecer sinal

DF37: 20 21 DC	165	JSR SHFTB	Deslocar
DF38: 84 A4	166	STY FPGEM	Y=0
DF3C: 60	167	RTS	
	168		
DF3D: A5 90	169	INT LDA FAC	
DF3F: C9 A0	170	CMP #540	(2°31?)
DF41: 80 20	171	BCC RTN12	Maio, sair
DF43: 20 DC DF	172	JSR GINT	
DF46: 84 AC	173	STY EXTRAFAC	Y=0
DF48: A5 A2	174	LDA FACSEN	
DF4A: 84 A2	175	STY FACSEN	
DF4C: A9 80	176	EOR #520	Testar sinal
DF4E: 2A	177	RDL	Guardar como status do carry
DF4F: A9 A0	178	LDA #540	Prep exp inicial de 2°31
DF51: B5 90	179	STA FAC	
DF53: A5 A1	180	LDA FAC+4	Guardar digito mais signif
DF55: B5 00	181	STA CHARMAC	Para EXP e para teste de
DF57: AC A3 08	182	JMP GSNIF	paridade em FPAR1
	183		
DF58: B5 9E	184	ZFAC STA FAC+1	Rutina INT precisa de todos
DF5C: B5 9F	185	STA FAC+2	os bytes = 0
DF5E: B5 A0	186	STA FAC+3	
DF60: B5 A1	187	STA FAC+4	
DF62: A8	188	TAY	Y=0
DF63: 60	189	RTN12 RTS	
	190		
	191	Mvalizar numero ponto flutuante em TXTPTR	
	192		
DF64: A0 00	193	FIN LDY #500	
DF66: A2 0A	194	LDX #50A	Gerar de TXPEXP a SERLEN
DF68: 94 99	195	FIN2 STY TXPEXP,X	(599 - 5A3)
DF6A: CA	196	DEX	
DF6B: 10 F8	197	BPL FIN2	
DF6D: 90 DF	198	BCC NOODSET	
DF6F: C9 20	199	CMP #'-'	
DF71: 00 04	100	BNE FIN3	
DF73: B6 A3	101	STX SERLEN	Indicar numero negativo se '-'
DF75: F0 04	102	BEA EVAL	
DF77: C9 28	103	FIN3 CMP #'+'	
DF79: 00 05	104	BNE CHOMP	
DF7B: 20 81 00	105	EVAL JSR CHRGET	
DF7E: 90 58	106	NOODSET BCC INSRTOIG	
DF80: C9 2E	107	CHOMP CMP #'.'	
DF82: F0 2E	108	BEA SETDP	
DF84: C9 A5	109	CMP #'E'	
DF86: 00 30	110	BNE ADJEXP	
DF88: 20 81 00	111	JSR CHRGET	Obter expoente
DF8B: 90 17	112	BCC GAGEX	
DF8D: C9 C9	113	CMP Minus	Expoente negativo?
DF8F: F0 0E	114	BEA SETSGN	Sim, indicar
DF91: C9 20	115	CMP #'-'	Maio pode estar em
DF93: F0 0A	116	BEA SETSGN	forma de TOXEN
DF95: C9 C8	117	CMP #plus	Semelhante para '+'
DF97: F0 0B	118	BEA DPDIG	
DF99: C9 28	119	CMP #'+'	
DF9B: F0 04	120	BEA DPDIG	
DF9D: 00 07	121	BNE SERCHK	Numero completo

		1122			
DFYF: 66 9C	1123	SETSON	ROR	EXPSON	;Indicar expoente negativo
DFAI: 20 B1 00	1124	OPDIO	JSR	CHRGF	;Obter prox algar do expoente
DFAA: 90 5C	1125	SGGEX	RCC	DETERP	;Numero, desviar
DFAG: 24 9C	1126	SGMCH	BIT	EXPSON	
DFAD: 10 0E	1127		BPL	ADJEXP	
DFAE: A7 00	1128		LDA	MSCO	
DFAC: 30	1129		SEC		;Tornar expoente negativo
DFAD: E5 9A	1130		SBC	EXPAN	
DFAF: 4C BA 0F	1131		JMP	ANEX	
DFE2: 66 9B	1132	SETOP	ROR	OPFLG	
DFBA: 24 9B	1133		BIT	OPFLG	
DFB6: 50 C3	1134		BVC	EVAL	;Desviar se proximo "+"
	1135				
DFEB: A5 9A	1136	ADJEXP	LDA	EXPON	;Ajustar expoente e sair
DFBA: 30	1137	ANEX	SEC		
DFBB: E5 99	1138		SBC	THREXP	
DFBD: E5 9A	1139		STA	EXPON	
DFBF: F0 52	1140		BEQ	EVDONE	
DFC1: 10 09	1141		BPL	OPRGHT	
DFC3: 20 6F 00	1142	OPLEFT	JSR	DIVIO	
DFC6: E6 9A	1143		INC	EXPON	
DFCB: 00 F9	1144		BNE	OPLEFT	
DFCA: F0 07	1145		BEQ	EVDONE	
DFCC: 20 53 00	1146	OPRGHT	JSR	MULIO	
DFCF: C6 9A	1147		DEC	EXPON	
DFD1: 00 F9	1148		BNE	OPRGHT	
DFD3: A5 A3	1149	EVDONE	LDA	SERLEN	;Negativo?
DFD5: 30 D1	1150		BMI	EVO	
DFD7: 60	1151		RTS		
	1152				
DFD8: 4C EA E1	1153	EVO	JMP	NEGOF	
	1154				
DFD8: 40	1155	INSRTDIO	PHA		;Guardar digito
DFDC: 24 9B	1156		BIT	OPFLG	;Aparece ponto decimal?
DFDE: 10 02	1157		BPL	NOP	;Nao, desviar
DFED: E6 99	1158		INC	THREXP	;Sim, ajustar
DFE2: 20 53 00	1159	NOP	JSR	MULIO	;Passar por cima do pto decimal
DFE3: A0	1160		PLA		;Ajustar digito a esquerda
DFE6: 30	1161		SEC		;do ponto decimal
DFE7: E9 30	1162		SBC	#'0'	;limpaca
DFE9: 20 EF 0F	1163		JSR	ADDACC	
DFEC: 4C 7B 0F	1164		JMP	EVAL	;Loop ate terminar
	1165				
	1166	Rotina para adicionar A a FAC			
	1167				
DFEF: 40	1168	ADDACC	PHA		
DFF0: 20 7D DE	1169		JSR	NOVNF	;Copiar FAC em ANF
DFF3: A0	1170		PLA		
DFF4: 20 A0 0E	1171		JSR	FLONT	
DFF7: A5 AA	1172		LDA	ANSSCH	
DFF9: 45 A2	1173		EOR	FACSON	
DFFB: 85 A0	1174		STA	SGMCH	
DFFD: A6 90	1175		LDX	FAC	;Mo sinal se FAC = 0
DFFF: 4C DB DA	1176		JMP	FADDT	
	1177				
EG02: A5 9A	1178	GETEXP	LDA	EXPON	;Mostrar expoente > 99?


```

E004: C9 06 1179      CMP    #50A
E004: 90 09 1180      BCC    #V0G      ;Mao, desviar
E003: A9 64 1181      LDA    #54A      ;Grande demais
E004: 24 9C 1182      BIT    EXP50N      ;Expoente negativo?
E00C: 30 11 1183      BNE    STEX      ;Sim, obteremos 0
E00E: 4C E7 08 1184      JMP    OVERFLOW      ;Mao, overflow
E011: 0A 11 1185      ASL     #V0E      ;Expoente velho vezes 10
E012: 0A 1186      ASL
E013: 10 1187      CLC
E014: 65 9A 1188      ADC    EXP0N
E016: 0A 1189      ASL
E017: 10 1190      CLC
E018: A0 00 1191      LOY    #500
E01A: 71 88 1192      ADC    (TXTPTR),Y      ;Somar novo algoritmo
E01C: 30 1193      SEC
E01D: E9 30 1194      SBC    #10      ;Compensar ASCII
E01F: 85 9A 1195      STX    STA EXP0N
E021: 4C A1 0F 1196      JMP    OP010
      1197
      33C      PUT    PARTE10,01
      11
E024: 96 3E 0C 12      CEM#ade HEX 963EBC5FFD ;99.999.999,9
E027: 5F F0
E029: 9E 6E 68 13      BIT#aeUH HEX 9E6E6B27FD ;999.999.999
E02C: 27 F0
E02E: 9E 6E 68 14      BIT#ao HEX 9E6E6B280D ;1.000.000.000
E031: 28 00
E033: A9 47 15      INPRY    LDA    #INMSG      ;Imprimir "IN"
E035: A0 C6 16      LOY    #1AMSG
E037: 20 4B E0 17      JSR    PRSTR
E03A: A5 76 18      LDA    CURLTN+1
E03C: A6 75 19      LOY    CURLTN
E03E: 85 9E 110      LHMRT    STA    FAC+1      ;Imprimir A,X em decimal
E040: 86 9F 111      STX    FAC+2
E042: A2 90 112      LOX    #590
E044: 38 113      SEC
E045: 20 8A 0E 114      JSR    FLO2
E048: 20 4E E0 115      PRINTFAC JSR    FOUT      ;Imprimir sua pto flut em FAC
E049: 4C 5F CE 116      PRSTR    JMP    STROUT      ;Imprimir string de A,X
      117
      118      *Converte FAC a string em STACK e aponta A,Y a ele
      119
E04E: A0 01 120      FOUT    LOY    #501
      121
      122      *Entrada de STR5 coloca string em 5FF (Y=0) de maneira
      123      *a forçar movimento do string a area de string
      124
E050: A7 20 125      FACSTRNG LDA    #'-'
E052: 88 126      OEY
E053: 24 A2 127      BIT    FAC50N
E055: 10 04 128      BPL    SFS6
E057: C8 129      BNY
E058: 99 FF 00 130      STA    STACK-1,Y
E058: 85 A2 131      SFS6    STA    FAC50N      ;Valor absoluto
E059: 84 A0 132      STY    STRNG2
E05F: C8 133      BNY
E060: A7 30 134      LDA    #530

```

ED62:	A6 9D	135		LDX	EAC	;Numero = 0?
ED64:	00 03	136		BNE	NOT2E	
ED66:	4C 71 E1	137		JMP	WNOUP	;Sim, terminar
ED68:	A9 00	138	NOT2E	LDA	#500	
ED6E:	ED 80	139		CPX	#500	;Numero >= 1?
ED6D:	FD 02	140		BEG	H3	
ED6F:	80 09	141		BCC	STE	;Sim, desviar
ED71:	A9 2E	142	H3	LDA	#BILHÃO	
ED73:	A0 ED	143		LDY	#BILHÃO	
ED75:	20 99 0C	144		JSR	FNULT	;Mover pto decimal e tornar o
ED78:	A9 F7	145		LDA	#5F7	... pto fixo para acelerar
ED7A:	85 99	146	STE	STA	TAPEXP	
ED7C:	A9 29	147	CHPB41	LDA	#BILHÃO	
ED7E:	A0 ED	148		LDY	#BILHÃO	
ED80:	20 CC DE	149		JSR	EDCMP	;Normalizar entre 100.000,000 e
ED83:	FD 1E	150		BEG	INTPART	999,999,999
ED85:	10 12	151		BPL	J010	
ED87:	A9 24	152	CHPB41	LDA	#CCENhede	
ED89:	A0 ED	153		LDY	#CCENhede	
ED8B:	20 CC DE	154		JSR	FCOMP	
ED8E:	FD 02	155		BEG	J010	
ED90:	10 DE	156		BPL	ROUH	
ED92:	20 53 00	157	J010	JSR	MULD	;Desv. se estiver no intervalo
ED95:	C6 99	158		BCC	TAPEXP	
ED97:	00 DE	159		BNE	CHPB41	
ED99:	20 AF 00	160	J010	JSR	DIVID	
ED9C:	E6 99	161		TWC	TAPEXP	
ED9E:	00 0C	162		BNE	CHPB41	
EDA0:	20 BA 04	163	ROUH	JSR	FAD2H	;Arredondar
EDA3:	20 0C 0F	164	INTPART	JSR	RINT	;Converter forma normal a int
EDA5:	A2 01	165		LDR	#501	;pointer DP
EDA8:	A5 99	166		LDA	TAPEXP	
EDAA:	1E	167		CLC		
EDAC:	A9 0A	168		ADC	#50A	;Testar se numero < .01
EDAD:	30 09	169		BNE	DPLOC	;Desv se neces exp negativo
EDAF:	C9 08	170		CMF	#50B	;Test se numero > 999,999,999
EDB1:	80 06	171		BCC	DPL	;Desv. se neces exp positivo
EDB3:	A9 FF	172		ADC	#5FF	;Subtrair 1
EDB5:	AA	173		TAX		;Montar a DP
EDB6:	A9 02	174		LDA	#502	
EDB8:	38	175	DPLOC	SEC		;Calcular expoente correto
EDB9:	C9 02	176	DPL	SBC	#502	
EDBB:	85 9A	177		STA	EXPON	;0 se nao ha expoente
EDBD:	86 99	178		STX	TAPEXP	;Numero de digitos antes de DP
EDBF:	8A	179		TXA		
EDC0:	FD 02	180		BEG	PUTDP	
EDC2:	10 13	181		BPL	NNSTB	;Desviar se nao comecou com DP
EDC4:	AA AD	182	FUTOP	LDY	STRING2	
EDC6:	A9 2E	183		LDA	#','	
EDC8:	C8	184		INY		
EDCA:	99 FF 00	185		STA	STACK-1,Y	
EDCC:	8A	186		TXA		
EDCE:	ED 06	187		BEG	SVY	
EDCF:	A9 30	188		LDA	#'0'	
EDD1:	C8	189		INY		
EDD3:	99 FF 00	190		STA	STACK-1,Y	
EDD5:	84 AD	191	SVY	STY	STRING2	

E067: A0 00	>92	ARESTR	LDY #500	;lerar numero enquanto se
E0D9: A2 00	>93		LDX #980	cria string
E0D8: A5 A1	>94	MSLUP	LDA FAC+4	
E0D0: 08	>95		CLC	
E0DE: 79 86 E1	>96		ADC DECTBL+3.1	
E0E1: B5 A1	>97		STA FAC+4	
E0E3: A5 A0	>98		LDA FAC+3	
E0E5: 79 85 E1	>99		ADC DECTBL+2,Y	
E0E8: B5 A0	>100		STA FAC+3	
E0EA: A5 9F	>101		LDA FAC+2	
E0EC: 79 84 E1	>102		ADC DECTBL+1,Y	
E0EF: B5 9F	>103		STA FAC+2	
E0F1: A5 9E	>104		LDA FAC+1	
E0F3: 79 83 E1	>105		ADC DECTBL,Y	
E0F6: B5 9E	>106		STA FAC+1	
E0F8: EB	>107		INX	;Contar em X
E0F9: 80 04	>108		BCC PARITY?	;Continuar a somar ou subtrair
E0FB: 10 0E	>109		BPL MSLUP	se numero decimal positivo e
E0FD: 30 02	>110		RMI COUNTEO	carry ligado ou numero decimal
E0FF: 30 0A	>111	PARITY?	BMI MSLUP	negativo e carry desligado
E1D1: 84	>112	COUNTEO	TXA	
E1D2: 90 04	>113		BCC MANDIGIT	
E1D4: 4F FF	>114		EOR #5FF	;ajustar o contador para o
E1D6: 6F 0A	>115		ADC #50A	caso de numero decimal posit
E1D8: 4F 2F	>116	MANDIGIT	ADC #'0'-1	;Converter contador a
E1DA: C8	>117		INX	salgarismo ASCII
E1DB: C8	>118		INX	
E1DC: C8	>119		INX	
E1DD: C8	>120		INX	
E1DE: 84 83	>121		STY WARPNT	;Guardar pointer a DECTBL
E1E0: A4 A0	>122		LDY STRING2	;Obter pointer ao string
E1E2: C8	>123		INX	
E1E3: AA	>124		TAX	
E1E4: 2F 7F	>125		AND #57F	
E1E6: 9F FF 00	>126		STA STACK-1,Y	
E1E9: C6 9F	>127		DEC TMPEXP	;Deslocar ponto decimal
E1EB: 00 06	>128		BNE SAVY	
E1ED: A9 2E	>129		LDA #'.'	;Inseri-lo no local adequado
E1EF: C8	>130		INX	
E1F0: 9F FF 00	>131		STA STACK-1,Y	
E1F3: 84 A0	>132	SAVY	STY STRING2	;Guardar pointer do string
E1F5: A4 83	>133		LDY WARPNT	;Obter pointer DECTBL
E1F7: 8A	>134		TXA	
E1F9: 4F FF	>135		EOR #5FF	;Trocar sinal de X
E1FA: 2F 90	>136		AND #5B0	
E1FC: AA	>137		TAX	
E1FE: C0 24	>138		CPY #524	
E1FF: 00 AA	>139		BNE MSLUP	;Loop ate terminar
E1A1: A4 A0	>140		LDY STRING2	
E1A3: 8F FF 00	>141	MWRACK	LDA STACK-1,Y	
E1A6: 88	>142		DEY	
E1A7: C9 30	>143		CMF #'0'	;Suprimir zeros finais
E1A9: F0 F8	>144		BEQ MWRACK	
E1AB: C9 2E	>145		CMF #'.'	;Se terminar em 0F, recobri-lo
E1AD: F0 01	>146		BEQ NEEDED?	
E1AF: C8	>147		INX	
E1B0: A9 28	>148	NEEDED?	LDA #'.'	

```

E142: A6 96 1149      LDZ EXPON
E144: F0 2E 1150      BEQ MARKEND ;Desviar se nao ha expoente
E146: 10 00 1151      BPL PUTEX ;Desviar se expoente positivo
E148: A7 00 1152      LDA NSD0
E14A: 30 1153      SEC
E14B: E5 96 1154      SRC EXPON ;Torna-lo negativo
E14D: AA 1155      TAX
E14E: A9 20 1156      LDA M'-1
E150: 99 01 01 1157 PUTEX STA STACK+1,Y
E153: A9 45 1158      LDA M'E'
E155: 99 00 01 1159      STA STACK,Y
E15B: 0A 1160      TXA ;Expoente em A
E159: A2 2F 1161      LDZ M'D'-1 ;Usar X para contar expoente
E15B: 30 1162      SEC ;em ASCII
E15C: E8 1163      INXTX
E15D: E9 0A 1164      SRC NSDn ;Dividir por 10
E15F: 80 FB 1165      BCS MHATX
E161: 69 3A 1166      AOC M'D'+10 ;Ajustar resto
E163: 99 03 01 1167      STA STACK+3,Y ;= Parte baixa de exp em ASCII
E166: 0A 1168      TXA ;Obter quociente
E167: 99 02 01 1169      STA STACK+2,Y ;= Parte alta do exp em ASCII
E16A: A9 00 1170      LDA NSD0
E16C: 99 04 01 1171      STA STACK+4,Y ;Marcar fim
E16F: F0 00 1172      BEQ PNTSTX
E171: 99 FF 00 1173 MHGUP STA STACK-1,Y
E174: A9 00 1174 MARKEND LDA NSD0
E176: 99 00 01 1175      STA STACK,Y
E179: A9 00 1176      PNTSTX LDA #1STACK
E17B: A0 01 1177      LDY #1STACK
E17D: 60 1178      RTS
336 PUT PARTEIV_01
11
E17E: 00 00 00 12 HALF HEX 000000000
E181: 00 00
13
14 *Representacao em 32 bits hexa das potencias de 10
15
E183: FA 0A 1F 16 DECTBL HEX FADAFD0 ;-1000000000
E184: 00
E187: 00 9B 16 17 HEX 009B680 ;10000000
E18A: 80
E18C: FF F0 80 18 HEX FFF0800 ;-1000000
E18E: C0
E18F: 00 01 86 19 HEX 0001860 ;100000
E192: A0
E193: FF FF 00 1A HEX FFFF080 ;-10000
E196: F0
E197: 00 00 03 11 HEX 0000030 ;1000
E19A: E0
E19B: FF FF FF 12 HEX FFFFFFFC ;-100
E19E: 9C
E19F: 00 00 00 13 HEX 0000000A ;10
E1A2: 0A
E1A3: FF FF FF 14 HEX FFFFFFFF ;-1
E1A6: FF
115
116 TEND =

```

	117			
E1A7: 20 70 DE	118	SBR	JSR M0ANF	;Calcular como potencia de 1/2
E1AA: A9 7E	119		LDA MHALF	
E1AC: A0 E1	120		LDY MHALF	
E1AE: 29 13 DE	121		JSR M0VFA	;Colocar 1/2 em FAC
	122			
E1B1: F0 70	123	FFMRT	BEQ EXP	;ARG*FAC -> FAC
E1B3: A5 A5	124		LDA ARG	
E1B5: 00 03	125		BNE PM1	
E1B7: 4C 6A 08	126		JMP AtoFAC	;Fazer FAC=0 se ARG=0
	127			
E1BA: A2 0A	128	PM1	LDX MTEMP3	
E1BC: A8 00	129		LDY #00	
E1BE: 20 45 DE	130		JSR M0ANF	;Guardar em TEMP3
E1C1: A5 AA	131		LDA ARGSEM	
E1C3: 10 0F	132		BPL PM2	;Desviar se argumento positivo
E1C5: 20 30 DF	133		JSR INT	;Obter parte inteira do exp
E1CB: A9 0A	134		LDA MTEMP3	
E1CD: A0 00	135		LDY M000	
E1CE: 20 CC DE	136		JSR FC0MP	;E potencia inteira
E1CF: 00 03	137		BNE PM2	
E1D1: 98	138		TYA	;Sim, permitir argum negativo
E1D3: A4 05	139		LDY CHARAC	;Obter paridade (de INT)
E1D4: 20 4F DE	140	PM2	JSR MFA	;Mover argumento a FAC
E1D7: 98	141		TYA	;Bit menos signif so pode ser
E1D8: 40	142		PHA	; ligado por LDY CHARAC
E1D9: 20 58 DC	143		JSR LOG	;Obter LOG (argumento)
E1DC: A9 0A	144		LDA MTEMP3	
E1DE: A0 00	145		LDY M000	
E1E0: 20 99 DC	146		JSR FMULT	;Calcular exponencial)4LOG (arg)
E1E3: 20 29 E2	147		JSR EXP	;Elevar a potencia e
E1E4: 60	148		PLA	;Exponete era inteiro negativo?
E1E7: 4A	149		LSR	
E1E8: 90 0A	150		BCC RTN13	;Nao, retornar
E1EA: A5 90	151	NEGOP	LDA FAC	;Resultado zero?
E1EC: F0 0A	152		BEQ RTN13	;Sim, retornar
E1EE: A5 A2	153		LDA FACSEM	
E1FD: A9 FF	154		EOR M0FF	
E1F2: 85 A2	155		STA FACSEM	
E1F4: 60	156	RTN13	RTS	
	157			
	158			Nos coeficientes dos valores indicados aqui
	159			Nao ajustados para precisao
	160			
E1F5: 81 38 AA	161	LOGE	HEX 8138AA3829	;LOG (e) na base 2
E1FB: 38 29				
E1FA: 07	162	EXPSEM	BF8 7	;Indice do numero de coefic
E1FB: 71 34 50	163		HEX 714368E56	;((log(2)^71)/71)
E1FE: 3E 56				
E200: 74 16 7E	164		HEX 74167EB318	;((log(2)^61)/61)
E203: 83 18				
E205: 77 2F EE	165		HEX 772FEE3B5	;((log(2)^51)/51)
E208: E3 05				
E20A: 7A 10 84	166		HEX 7A10841C2A	;((log(2)^41)/41)
E20D: 1C 2A				
E20F: 7C 63 59	167		HEX 7CA359580A	;((log(2)^31)/31)
E212: 58 0A				

E214:	7E 75 F0 168	HEX	7E75F0E7C6 ;(log12)*21/21
E217:	E7 C6		
E219:	80 31 72 169	HEX	8031721810 ;log12)
E21C:	18 10		
E21E:	01 00 00 170	HEX	8100000000 ;1
E221:	00 00		
			171
E223:	A9 F5 172	EXP	LDA #LOGe ;e*FAC -> FAC
E225:	A0 E1 173		LDY #LOGe
E227:	20 99 DC 174		JSR FMULT ;Prep para calc e*(xLOG(e))
E22A:	A5 AC 175		LDA EXTRAFAC
E22C:	69 50 176		ADC #550
E22E:	90 03 177		BCC X1
E230:	20 94 DE 178		JSR R32ND
E233:	85 92 179	X1	STA EXTRAFAC
E235:	20 80 DE 180		JSR MAF ;Copiar a ARG
E238:	A5 90 181		LDA FAC
E23A:	C9 88 182		CMP #508 ;Dentro do intervalo
E23C:	90 03 183		BCC X2 ;Nao, desviar
E23E:	20 45 00 184	00F	JSR OUTOFRNG ;Zero ou overflow
E241:	20 30 0F 185	X2	JSR INT ;Obter parte inteira em FAC
E244:	A5 00 186		LDA CHARAC
E246:	18 187		CLC
E247:	69 81 188		ADC #581 ;Segundo teste de intervalo
E249:	F0 F3 189		BEQ 00E
E24B:	38 190		SEC
E24C:	E9 01 191		SBC #501
E24E:	48 192		PHA
E24F:	A2 05 193		LDR #505
E251:	85 A5 194	X3	LDA ARG,X ;Trocar ARG com FAC
E253:	84 99 195		LDY FAC,X
E255:	95 90 196		STA FAC,X
E257:	94 A5 197		STY ARG,X
E259:	CA 198		DEX
E25A:	10 F5 199		BPL X3
E25C:	A5 92 1100		LDA EXTRAFAC
E25E:	85 AC 1101		STA EXTRAFAC
E260:	20 C4 DA 1102		JSR FSUBT ;Subtrair parte inteira
E263:	20 EA E1 1103		JSR MEXP
E266:	A9 FA 1104		LDA MEXPSE
E268:	A0 E1 1105		LDY #EXPSE
E26A:	20 0C E2 1106		JSR SERIES ;Usar series para a parte frac
E26B:	A9 00 1107		LDA #500
E26F:	85 A0 1108		STA SGNCP1
E271:	68 1109		PLA
E272:	20 2A 00 1110		JSR ADEX2 ;Somar exp da parte inteira
E275:	60 1111		RTS
			1112
E276:	85 A0 1113	00B5E	STA SERPMT ;Calcular auxbx*3rea*5*...
E278:	84 AE 1114		STY SERPMT+1 ;Com SERPMT apontado aos
E27A:	20 38 DE 1115		JSR MOVF coeficientes ...,b,a.
E27D:	A9 93 1116		LDA RTENP1 ;Numero de coefic = SERLEN+1
E27F:	20 99 DC 1117		JSR FMULT ;x^2
E282:	20 90 E2 1118		JSR SERMAIN ;Fazer serie com x^2
E285:	A9 93 1119		LDA RTENP1 ;Obter x de novo
E287:	A0 00 1120		LDY #500
E289:	4C 99 DC 1121		JMP FMULT ;Multiplicar pela serie e sair

E28C:	85	A0	1122			
E28E:	84	AE	1123	SERIES	STA	SERPNT
E290:	20	38	9E	1125	SEMWATN	JSR
E293:	B1	A0	1126		LDN	(SERPNT),Y
E295:	85	A3	1127		STA	SERLEN
E297:	A4	A0	1128		LDY	SERPNT
E299:	C8		1129		INT	
E29A:	98		1130		TYA	
E29B:	00	02	1131		RNE	SS
E29D:	E6	AE	1132		INC	SERPNT+1
E29F:	85	A0	1133	SS	STA	SERPNT
E2A1:	A4	AE	1134		LDY	SERPNT+1
E2A3:	20	99	0C	1135	SERLOOP	JSR
E2A6:	A5	A0	1136		LDN	SERPNT
E2A8:	A4	AE	1137		LDY	SERPNT+1
E2AA:	18		1138		CLC	
E2AB:	69	05	1139		ADC	INSDS
E2AD:	90	01	1140		RCC	MXTERM
E2AF:	C8		1141		INT	
E2B0:	85	A0	1142	MXTERM	STA	SERPNT
E2B2:	84	AE	1143		STY	SERPNT+1
E2B4:	20	08	0A	1144	JSR	FADD
E2B7:	A9	90	1145		LDN	WTEMP2
E2B9:	A0	00	1146		LDY	INSDS
E2BA:	C6	A3	1147		DEC	SERLEN
E2BD:	00	E4	1148		RNE	SERLOOP
E2BF:	60		1149	RTNS4	RTS	
			1150			
E2C8:	98	35	44	1151	RND40J1	HEX
E2C3:	7A					
E2C4:	68	28	B1	1152	RND40J2	HEX
E2C7:	46					
			1153			
E2C8:	20	9C	0E	1154	RND	JSR
E2CB:	AA		1155		TAX	
E2CC:	30	18	1156		BMI	ROI
E2CE:	A9	C9	1157		LDN	WINDSEED
E2D0:	A0	00	1158		LDY	INSDS
E2D2:	20	13	0E	1159	JSR	NOVFN
E2D5:	BA		1160		TXA	
E2D6:	F0	E7	1161		REN	RTNS4
E2D8:	A9	C0	1162		LDN	WIND40J1
E2DA:	A0	E2	1163		LDY	WIND40J1
E2DC:	20	99	0C	1164	JSR	FNULT
E2DF:	A9	C4	1165		LDN	WIND40J2
E2E1:	A0	E2	1166		LDY	WIND40J2
E2E3:	20	08	0A	1167	JSR	FADD
E2E6:	A6	A1	1168	ROI	LDX	FAC+4
E2E8:	A5	9E	1169		LDN	FAC+1
E2EA:	85	A1	1170		STA	FAC+4
E2EC:	BA	9C	1171		STX	FAC+1
E2EE:	A9	00	1172		LDN	INSDS
E2F0:	85	A2	1173		STA	FACSEN
E2F2:	A5	9C	1174		LDN	FAC
E2F4:	85	AC	1175		STA	EXTRAFAC
E2F6:	A9	80	1176		LDN	INSDS

!Calcular $a+b+c+2r...$
!Com SERPNT apontado aos
coeficientes ...,b,a.
!Colocar em SERLEN valor do
inicio da tabela e apontar
!SERPNT ao ultimo coeficiente
!(que e o primeiro da tabela)

!Passar SERPNT ao prox coefic

!Somar proximo coeficiente
!Obter x de novo

!Loop ate terminar

!Obter sinal do argumento
!Testar
!Se -, usar FAC corrente

!Mover raiz corrente da FAC

!Restaurar sinal
!Sair se RND(0)
!Juntar

!Juntar de novo

!Mais
!Trocar byte mais signif com
byte menos signif

!Tomar valor absoluto
!Posicionar bit extra ao acaso
!ajustar para o intervalo 0-1

E2F8: 85 90	1177	STA	FAC	
E2FA: 20 48 08	1178	JSR	SIGNIF	;Normalizar
E2FD: A2 C9	1179	LDX	NRNDSEED	;Mover FAC a raiz RND
E2FF: A0 00	1180	LDY	MS00	
E301: 4C 45 0E	1181	JMP	MSWF	
	337	PUT	PORTE2a,02	
	11			
E304: A9 80	12	COB	LDA	MPIdiv2
E306: A0 E3	13		LDY	MPIdiv2
E308: 20 08 0A	14		JSR	FADD
	15			
E30B: 20 7D 0E	16	SIN	JSR	MSWF
E30E: A9 85	17		LDA	MP1vez2
E310: A0 E3	18		LDY	MP1vez2
E312: A6 AA	19		LDX	ARGSEM
E314: 20 78 00	110		JSR	DIY
E317: 20 7D 0E	111		JSR	MSWF
E31A: 20 3D 0F	112		JSR	INT
E31D: A9 80	113		LDA	MS00
E31F: 85 A8	114	STA	SEMCPR	
E321: 20 C4 0A	115	JSR	FSUBT	;Subtrair para obter mod (2pi)
E324: A9 8A	116	LDA	MSQUARTER	
E326: A0 E3	117		LDY	MSQUARTER
E328: 20 C1 0A	118	JSR	FSUB	;Conv argum ao prim quadrante
E32B: A5 A2	119	LDA	FACSEM	; intervalo 0 a 1/4 como
E32D: 48	120	PHA		; multiplos de 2 pi
E32E: 10 00	121	BPL	SI1	
E330: 20 BA 0A	122	JSR	FACON	
E333: A5 A2	123	LDA	FACSEM	
E335: 30 09	124	BNE	SI2	
E337: A5 16	125	LDA	SIGNFLG	
E339: A9 FF	126	EOB	MSFF	
E33E: 85 16	127	STA	SIGNFLG	
E340: 20 EA E1	128	SI1	JSR	NEGOP
E34D: A9 BA	129	SI2	LDA	MSQUARTER
E34E: A0 E3	130		LDY	MSQUARTER
E34H: 20 D8 0A	131	JSR	FADD	
E347: 68	132	PLA		
E34B: 10 00	133	BPL	SI3	
E34A: 20 EA E1	134	JSR	NEGOP	
E34C: A9 BF	135	SI3	LDA	MSMSER
E34F: A0 E3	136		LDY	MSMSER
E351: 4C 76 E2	137	JMP	ODDSER	
	138			
E354: 20 3E 0E	139	TAN	JSR	MSVIF
E357: A9 80	140		LDA	MS00
E359: 85 16	141	STA	SIGNFLG	
E35B: 20 08 E3	142	JSR	SIN	
E35E: A2 BA	143	LDX	HTEMP3	
E360: A0 00	144	LDY	MS00	;Guardar seno em TEMP3
E362: 20 01 E3	145	JSR	REC	
E365: A9 93	146	LDA	HTEMP1	
E367: A0 00	147	LDY	MS00	
E369: 20 13 0E	148	JSR	MSVFN	;Obter FAC e calcular cosseno
E36C: A9 80	149	LDA	MS00	
E36E: 85 A2	150	STA	FACSEM	
E370: A5 16	151	LDA	SIGNFLG	

E372:	20 7C E3	152	JSR	TAK2	
E375:	A9 8A	153	LDA	WTEMP3	;Deter sens
E377:	A0 00	154	LDY	#000	
E379:	4C 90 00	155	JMP	F01V	;E dividir
		156			
E37C:	4B	157	TAK2	PHA	
E37D:	4C 30 E3	158	JMP	S11	
		159			
E380:	B1 49 0F	160	Pldiv2	HEX	81490FDA2
E383:	DA A2				
E385:	B3 49 0F	161	P1veze2	HEX	83490FDA2
E388:	DA A2				
E38A:	7F 00 00	162	QUARTER	HEX	7F0000000
E38D:	00 00				
		163			
		164			
		165			
E38F:	05	166	SIMSER	0FB 5	;Indice do numero de coefic
E390:	84 E6 1A	167	HEX	84E61ACD18	; (2pi)*11/111
E393:	20 10				
E395:	86 20 07	168	HEX	862007F8F8	; (2pi)*9/91
E398:	F8 F8				
E39A:	87 9F 68	169	HEX	879F68E9D1	; (2pi)*7/71
E39D:	89 01				
E39F:	87 23 35	170	HEX	872335FE1	; (2pi)*5/51
E3A2:	0F E1				
E3A4:	B6 A5 50	171	HEX	B6A550E728	; (2pi)*3/31
E3A7:	E7 28				
E3A9:	B3 49 0F	172	HEX	83490FDA2	; 2pi
E3AC:	DA A2				
E3AE:	A6 03 C1	173	HEX	A603C1C804	
E3B1:	C8 D4				
E3B3:	C8 05 DA	174	HEX	C805DACECA	
E3B6:	CE DA				
		175			
E3B8:	A5 A2	176	ATM	LDA	FACSGN
E3BA:	4B	177		PHA	
E3BC:	10 03	178		BPL	ATM1
E3BD:	20 EA E1	179		JSR	HCBOP
E3C0:	A5 90	180	ATM1	LDA	FAC
E3C2:	4B	181		PHA	
E3C3:	C9 81	182		CHP	W81
E3C5:	90 07	183		BCC	ATM2
E3C7:	A9 20	184		LDA	BHUM
E3C9:	A0 DC	185		LDY	BHUM
E3CB:	20 80 00	186		JSR	F01V
E3CE:	A9 EB	187	ATM2	LDA	BATNSER
E3D0:	A0 E3	188		LDY	BATNSER
E3D2:	20 76 E2	189		JSR	000SER
E3D5:	68	190		PLA	
E3D6:	C9 81	191		CHP	W81
E3D8:	90 07	192		BCC	ATM3
E3DA:	A9 80	193		LDA	WIdiv2
E3DC:	A0 E3	194		LDY	WIdiv2
E3DE:	20 C1 DA	195		JSR	FSUB
E3E1:	68	196	ATM3	PLA	
E3E2:	10 03	197		BPL	RTM15

;Compensar para normalizacao

```

E3E4: 4C EA E1 :99      JMP  NEGOP
E3E7: 60          :99      RTNLS  RTS
                        :100
E3E8: 08          :101      ATNSR  DFB  $00      ;Índice do número de coefic
E3E9: 76 83 83 :102      HEX  7683830003
E3EC: 80 08
E3ED: 79 1E F4 :103      HEX  791EF4AF5
E3F1: A4 F5
E3F3: 78 95 FC :104      HEX  7883FC8010
E3F6: 80 10
E3F8: 7C 0C 1F :105      HEX  7C0C1F67CA
E3FB: 67 CA
E3FD: 7C 0E 53 :106      HEX  7C0E538BC1
E400: C8 C1
E402: 7D 14 6A :107      HEX  7D146A7D4C
E405: 7D 4C
E407: 7D 87 EA :108      HEX  7D87EAS17A
E40A: 51 7A
E40C: 7D 63 3D :109      HEX  7D633D887E
E40F: 88 7E
E411: 7E 92 44 :110      HEX  7E9244993A
E414: 99 3A
E416: 7E 4C CC :111      HEX  7E4CCC91C7
E419: 91 C7
E41B: 7F A4 AA :112      HEX  7FAAAA8A3
E41E: AA 13
E420: 81 00 00 :113      HEX  8100000000
E423: 00 00
                        :114
                        :115      *Rotina CMDET (e raíz RMD) a serem colocados em
                        :116      *SR1 na página zero
                        :117
E425: E6 88      :118      ZPSTUFF INC  TXPTR
E427: 00 02      :119      BNE  NID
E429: E6 89      :120      INC  TXPTR+1
E42B: AD 7A 00 :121      NID   LDA  ENDIMP
E42E: C9 3A      :122      CMP  #' '      ;Set com carry lig se não numer
E430: 8D DA      :123      BCS  RTNLS      ;Ligar flag Z se "1"
                        :124      *      ;os fim de linha
E432: C9 2D      :125      CMP  #' '      ;Saltar espaços
E434: F0 EF      :126      BEQ  ZPSTUFF
E436: 38          :127      SEC
E437: E9 3D      :128      SBC  '0'      ;Desl carry se numer, ligar se
E439: 38          :129      SEC      ; não numer e conserv valor de A
E43A: E9 0D      :130      SBC  $100-'0'
E43C: 6D          :131      RTNLS  RTS
                        :132
E43D: 8D 4F C7 :133      HEX  8D4FC75258 ;Semente para número aleatório
E440: 52 58
                        :134
E442: A2 FF      :135      COLDET LDR  WFF
E444: BA 76      :136      STX  CMLIN+1 ;Inicializar modo direto
E446: A2 FB      :137      LDX  WFB      ;E pilha
E448: 9A          :138      TXS      ;Os 4 bytes mais altos da pilha
E449: A9 42      :139      LDA  MCOLDST ;são usados para pointer e
E44B: AD E4      :140      LDY  B7COLDET ; número da linha para input
                        :141      *      ; de linha

```

E44D: 85 01	1142	STA	GOWAH+1	
E44F: 84 02	1143	STY	GOWAH+2	
E451: 85 04	1144	STA	GOSTROUT+1	
E453: 84 05	1145	STY	GOSTROUT+2	
E455: 20 0A E5	1146	JSR	WORNAL	;Iniciar texto normal
E45B: A9 AC	1147	LDA	RS4C	;Preparar areas de jump
E45A: 85 00	1148	STA	GOWAH	
E45C: 85 03	1149	STA	GOSTROUT	
E45E: 85 90	1150	STA	JMPADS	
E460: 85 0A	1151	STA	USR	;Inicializar enderecos USR
E462: A9 B3	1152	LDA	R16ERR	; com rotina de erro
E464: A0 04	1153	LDY	R10ERR	
E466: 85 08	1154	STA	USR+1	
E468: 84 0C	1155	STY	USR+2	; "Valor ilegal"
E46A: A2 0C	1156	LOX	MS1C	
E46C: 80 24 E4	1157	LDA	ZPSTUFF-1,X	
E46F: 95 88	1158	STA	CHGET-1,X	
E471: 86 F1	1159	STX	SPEEDZ	;Inicializar SPEED com
E473: CA	1160	DEX		;ZSS (SPEEDZ-1)
E474: 00 F6	1161	BNE	WVZF	
E476: 86 F2	1162	STX	TRCFLG	;ligar NOTRACE
E478: 8A	1163	TXA		
E479: 85 A4	1164	STA	FPGEN	;Contem 0 exceto na rotina INT
E47B: 85 54	1165	STA	LASTPT+1	;Sempre contem 0
E47D: 48	1166	PHA		
E47E: A9 03	1167	LDA	MS03	;Inicializar OSOLEN com valor
E480: 85 BF	1168	STA	OSOLEN	esperado por GABAB
E482: 20 20 CE	1169	JSR	CR00	
E485: A9 01	1170	LDA	MS01	;Prep pointer fantasia a \$101
E487: 80 FD 01	1171	STA	TM-3	
E48A: 80 FC 01	1172	STA	TM-4	
E48D: A2 55	1173	LOX	MS35	;Inicializar indice aos
E48F: 86 52	1174	STX	TEMPPT	descriptores tempor. de STRAMES
E491: A9 01	1175	LDA	MS01	;teste de area de memoria
E493: 80 FF 07	1176	STA	R16K	
E495: A9 02	1177	LDA	MS02	
E498: 80 FF 47	1178	STA	ESPW16K	
E49B: C0 FF 07	1179	CMF	R16K	
E49E: 00 09	1180	BNE	T48K	
E4A0: A9 00	1181	LDA	MS00	
E4A2: 80 40 04	1182	STA	FLGRAM	
E4A5: A0 1F	1183	LDY	MS1F	
E4A7: C0 28	1184	BNE	STHINEX	
E4A9: A9 03	1185	LDA	MS03	
E4AB: 80 FF 87	1186	STA	R48K	
E4AE: 40 FF 87	1187	EDB	R48K	
E4B1: F0 0E	1188	BEB	SET48K	
E4B3: A9 01	1189	LDA	MS01	
E4B5: 80 40 04	1190	STA	FLGRAM	
E4B8: A0 35 04	1191	LDA	TSUTUL	
E4BB: 00 12	1192	BNE	HS5FFF	
E4BD: A0 5F	1193	LDY	MS5F	
E4BF: 00 10	1194	BNE	STHINEX	
E4C1: A9 02	1195	LDA	MS02	
E4C3: 80 40 04	1196	STA	FLGRAM	
E4C6: A0 35 04	1197	LDA	TSUTUL	
E4C9: 00 04	1198	BNE	HS5FFF	

```

E4CB: A0 9F 1199      LOY  #99F
E4CD: 00 02 1200      BNE  STINENH
E4CF: A0 5F 1201      #05FFF LOY  #55F
E4D1: A9 FF 1202      STINENH LDA  #5FF
E4D3: 05 73 1203      STA  #ENHSIZ
E4D5: 04 74 1204      STY  #ENHSIZ+1
E4D7: 05 6F 1205      STA  FRETOP
E4D9: 04 70 1206      STY  FRETOP+1
E4DB: 05 71 1207      STA  FRESPC
E4DD: 04 72 1208      STY  FRESPC+1
E4DF: A5 47 1209      LDA  PVAL
E4E1: 00 04 1210      BNE  DIV1000
E4E3: A2 08 1211      LDX  #808
E4E5: 00 02 1212      BNE  DIVBAS
E4E7: A2 10 1213      DIV1000 LDX  #510
E4E9: A0 00 1214      DIVBAS LOY  #500
E4EB: 04 67 1215      STY  BASLOC
E4ED: 06 68 1216      STX  BASLOC+1
E4EF: 04 06 1217      STY  LOCK      ;inicializar byte LOCK e
E4F1: 98 1218      TYA
E4F2: 91 67 1219      STA  (BASLOC),Y ;byte inicial do programa
E4F4: E6 67 1220      INC  BASLOC
E4F6: 00 02 1221      BNE  NOBASLOC
E4F8: E6 68 1222      INC  BASLOC+1
E4FA: A5 67 1223      NOBASLOC LDA  BASLOC
E4FC: A4 60 1224      LOY  BASLOC+1
E4FE: 20 02 C4 1225      JSR  BEASON
E501: 20 62 C9 1226      JSR  SCOTCH
E504: A0 35 04 1227      LDA  TSUTBL
E507: F0 11 1228      BFR  DIVBAS3
E509: A0 F0 9F 1229      LDA  TSCMT
E50C: C9 01 1230      CMP  #501
E50E: 00 0A 1231      BNE  INVBAS3
E510: A0 F1 9F 1232      LDA  INTCMT
E513: A6 F2 9F 1233      LDX  INTCMT+1
E516: 05 67 1234      STA  BASLOC
E518: 06 68 1235      STX  BASLOC+1
E51A: A9 03 1236      INVBAS3 LDA  #503
E51C: 05 AF 1237      STA  BASLOC+1
E51E: A0 68 00 1238      LDA  BASLOC+1
E521: 05 80 1239      STA  BASLOC1
E523: A9 5F 1240      LDA  #STROUT
          PUT  PARTE28,02
          338
          31
          32      ;Agora recobrir areas 0-5
          33
E525: A0 CE 1244      LOY  #1STROUT
E527: 05 04 1245      STA  #STROUT+1
E529: 04 05 1246      STY  #STROUT+2
E52B: A9 28 1247      LDA  #NESTART
E52D: A0 C7 1248      LOY  #1NESTART
E52F: 05 01 1249      STA  #GOWARN+1
E531: 04 02 1250      STY  #GOWARN+2
E533: 6C 01 00 1251      JMP  (GOWARN+1)
          312
          313      ;*****
          314      ;

```

```

>15 * Graficos,etc *
>16 * *
>17 =====
>18
ES36: 20 BF 00 >19 CALL JSR FRMMUN
ES39: 20 4C 0A >20 JSR GETADR
ES3C: 4C 50 00 >21 JMP (LDMUN)
>22
>23 *Observar que CALL nao prepara os registradores
>24 *Em um CALL:
>25 * Y tem o byte de baixa ordem do endereço CALL
>26 * A tem o byte de alta ordem do endereço CALL
>27 * X contem $70
>28
ES3F: 40 >29 RTS
>30
>31 *Subrotinas baixa resolucao
>32 *Subrotina para obter coordenadas *,*
>33
ES40: 20 12 0A >34 PLOTFMS JSR GETBYT ;Obter primeira coordenada
ES43: E0 30 >35 CPX #30 ;Testar se menor que 40
ES46: 80 13 >36 BCS GOERR
ES49: 86 F0 >37 STX FIRST
ES4C: A9 2C >38 LDA #' '
ES4F: 20 E8 01 >39 JSR SYNCHR ;Teste de sintaxe
ES52: 20 12 0A >40 JSR GETBYT ;Obter segunda coordenada
ES55: E0 30 >41 CPX #30 ;Tem que ser < 40
ES58: 80 05 >42 BCS GOERR
ES5B: 86 2C >43 STX #2 ;Preparar #2 e #2
ES5E: 86 20 >44 STX #2
ES61: 40 >45 RTS
>46
ES6A: 4C E3 04 >47 GOERR JMP IGERR
>48
ES6D: 20 40 E5 >49 LINC00R JSR PLOTFMS ;Obter coordenadas A,B
ES6F: E4 F0 >50 CPX FIRST ;B/A?
ES72: 80 06 >51 BCS AT? ;Nao, desviar
ES75: A5 F0 >52 LDA FIRST ;Sim, trocar A por B
ES78: 85 2C >53 STA #2
ES7B: 85 20 >54 STA #2
ES7E: 86 F0 >55 STX FIRST
ES81: A9 C5 >56 AT? LDA #at
ES84: 20 E8 01 >57 JSR SYNCHR ;Teste de sintaxe
ES87: 20 12 0A >58 JSR GETBYT
ES8A: E0 30 >59 CPX #30 ;Testar se < 40
ES8D: 80 E2 >60 BCS GOERR
ES90: 40 >61 RTS
>62
ES93: 20 40 E5 >63 PLOT JSR PLOTFMS
ES96: 8A >64 TGA
ES99: A4 F0 >65 LDY FIRST
ES9C: C0 2B >66 CPY #2B ;Coordenada X tem que ser < 40
ES9F: 80 07 >67 BCS GOERR
ESA2: 4C F0 >68 JMP PLOT2
>69
ESA5: 20 50 E5 >70 HLTH JSR LINC00R
ESA8: 8A >71 TGA

```

ESBA: A4 2C	172	LDI	R2	
ESBC: C0 2B	173	CPY	R52B	;Coordenada X tem que ser (40
ESBE: B0 CA	174	RCS	GOERR	
ESBD: A4 FD	175	LDI	FIRST	
ESBE: 4C 19 FB	176	JMP	RLINE	;Ir a rotina do monitor
	177			
ESBF: 20 5D E5	178	VLIN	JSR	LINCOOR
ESBG: BA	179		TXA	
ESBH: AB	180		TAY	
ESBA: C0 2B	181	CPY	R52B	;Coordenada X tem que ser (40
ESBC: B0 BC	182	RCS	GOERR	
ESBE: A5 FD	183	LOA	FIRST	
ESBD: 4C 2B FB	184	JMP	RLINE	;Ir a rotina do monitor
	185			
ESAG: C9 00	186	COLOR=	CHP	R500
ESAS: 00 03	187	BHE	EVALUAR	
ESAT: 20 B1 00	188	JSR	CHRGST	
	189			
	190			*Rotinas nao graficas
	191			
ESAA: 20 12 BA	192	EVALUAR	JSR	GETBYT
ESAD: BA	193		TXA	
ESAE: 29 0F	194	AND	R50F	
ESAF: AA	195		TAX	
ESBI: 29 03	196	AND	R503	
ESBD: 00 04	197	BHE	STCON	
ESBE: BA	198		TXA	
ESBF: AA	199		LSR	
ESBG: AA	1100		LSR	
ESBH: AA	1101		TAX	
ESBI: BA	1102	STCON	TXA	
ESBA: 4C 44 FB	1103	JMP	SETCOL	
ESBD: 20 12 BA	1104	VLIN	JSR	GETBYT
ESBE: CA	1105		OEZ	
ESBF: BA	1106		TXA	
ESBG: C9 18	1107	CHP	R518	;Intervalo correto?
ESBH: B0 94	1108	RCS	GOERR	
ESBI: 4C 5C FB	1109	JMP	TABU	;Usar rotina do monitor
	1110			
ESBF: 20 12 BA	1111	SPEED	JSR	GETBYT
ESBG: BA	1112		TXA	;1000 - SPEED -> SPEED2
ESBH: 49 FF	1113	EBR	WFF	
ESBI: AA	1114		TAX	
ESBD: EB	1115		IMX	
ESBE: BA F5	1116	STX	SPEED2	
ESBF: 40	1117		RTS	
	1118			
ESBA: 30	1119	TRACE	SEC	
ESBG: 90	1120		HEX	90
ESBH: 18	1121	NOTRACE	CLC	
ESBI: 66 F2	1122		ROR	TROFLG
ESBD: 60	1123		RTS	
	1124			
ESBA: A7 00	1125	NORMAL	LOA	R500
ESBC: 2C	1126		HEX	2C
ESBD: A7 7F	1127	INVERSE	LOA	R57F
ESBE: A2 00	1128	NNN	LOX	R500

```

E5E1: 85 32 1129 SI STA INWFLG
E5E3: 86 13 1130 STX ORWASE
E5E5: 40 1131 RTS
1132
E5E6: 40 1133 RTS
1134
E5E7: 20 BF D0 1135 HINEN JSR FIRMUN ;Obter HINEN especificado
E5EA: 20 6C DA 1136 JSR GETADR
E5ED: A5 50 1137 LDA LINMUN
E5EF: C5 40 1138 CNP STREND
E5F1: A5 51 1139 LDA LINMUN+1
E5F3: E5 6E 1140 SRC STREND+1
E5F5: 80 03 1141 ECS SETHI ;Acima da tabela de variaveis?
E5F7: 4C FF C6 1142 JMH JMP NENORN ;Nao erro
1143
E5FA: A5 50 1144 SETHI LDA LINMUN
E5FC: 85 73 1145 STA NENSI2
E5FE: 85 6F 1146 STA FRETOP
E600: A5 51 1147 LDA LINMUN+1
E602: 85 74 1148 STA NENSI2+1
E604: 85 70 1149 STA FRETOP+1
E606: 40 1150 RTS
329 PUT PARTE2C,02
11
E607: 20 BF D0 12 LONEN JSR FIRMUN
E60A: 20 6C DA 13 JSR GETADR
E60D: A5 50 14 LDA LINMUN
E60F: C5 73 15 CNP NENSI2
E611: A5 51 16 LDA LINMUN+1
E613: E5 74 17 SRC NENSI2+1
E615: 80 03 18 ECS JMH
E617: A5 50 19 LDA LINMUN
E619: C5 49 110 CNP VARTAB
E61B: A5 51 111 LDA LINMUN+1
E61D: E5 6A 112 SRC VARTAB+1
E61F: 90 06 113 SCC JMH
E621: A5 50 114 LDA LINMUN
E623: 85 49 115 STA VARTAB
E625: A5 51 116 LDA LINMUN+1
E627: 85 6A 117 STA VARTAB+1
E629: 4C 83 C9 118 JMP CLEARC ;LONEN limpa variaveis
119
E62C: A9 40 120 ONERR LDA Nqoto
E62E: 20 E8 D1 121 JSR SYNCHR
E630: A5 88 122 LDA TXTPTR
E632: 85 F4 123 STA TXTPSV
E635: A5 89 124 LDA TXTPTR+1
E637: 85 F5 125 STA TXTPSV+1
E639: 38 126 SEC
E63A: 66 08 127 ROR ORBFLG
E63C: A5 75 128 LDA CURLIN
E63E: 85 F6 129 STA CURLSV
E640: A5 76 130 LDA CURLIN+1
E642: 85 F7 131 STA CURLSV+1
E644: 20 D1 CC 132 JSR REPN ;Ignorar resto da linha
E647: 4C 80 CC 133 JMP ADDCOM
134

```

	135	Rotina de tratamento de erro se ONERR GOTO estiver ativo		
	136			
E644: 06 0E	137	INWOLERR STX	ERRMM	
E64C: A6 F0	138	LOX	REMSY	Get pointer da pilha
	139	*		guardando em MEMST
E64E: 06 0F	140	STX	ERRST	testar
E650: A5 75	141	LDA	CURLIN	
E652: 05 0A	142	STA	ERRLIN	
E654: A5 76	143	LDA	CURLIN+1	
E656: 05 0B	144	STA	ERRLIN+1	
E658: A5 79	145	LDA	OLDTEXT	
E65A: 05 0C	146	STA	ERRPOS	
E65C: A5 7A	147	LDA	OLDTEXT+1	
E65E: 05 0D	148	STA	ERRPOS+1	
E660: A5 F4	149	LDA	TXTPSR	
E662: 05 0B	150	STA	TXTPTR	
E664: A5 F5	151	LDA	TXTPSR+1	
E666: 05 09	152	STA	TXTPTR+1	
E668: A5 F6	153	LDA	CURLSV	
E66A: 05 75	154	STA	CURLIN	
E66C: A5 F7	155	LDA	CURLSV+1	
E66E: 05 76	156	STA	CURLIN+1	
E670: 20 07 00	157	JSR	CHRGOT	
E672: 20 03 0C	158	JSR	GOTO	Ir a linha ONERR especificada
E674: 4C EC CA	159	JMP	MEMST	
	160			
E679: A5 0A	161	RESUME	LDA	ERRLIN
E67B: 05 75	162		STA	CURLIN
E67D: A5 0B	163		LDA	ERRLIN+1
E67F: 05 76	164		STA	CURLIN+1
E681: A5 0C	165		LDA	ERRPOS
E683: 05 0B	166		STA	TXTPTR
E685: A5 0D	167		LDA	ERRPOS+1
E687: 05 09	168		STA	TXTPTR+1
E689: A6 0F	169		LOX	ERRSTX
E68B: 9A	170		TXS	Restaurar pointer da pilha ao
E68C: 4C EC CA	171		JMP	valor que tinha antes do erro
	172			
E68F: 4C F1 01	173	JSYN	JMP	SYNERR
	174			
E692: 00 F0	175	DEL	RCS	JSYN
E694: A6 AF	176		LOX	PRGEND
E696: 06 09	177		STX	WRTAM
E698: A6 00	178		LOX	PRGEND+1
E69A: 06 0A	179		STX	WRTAM+1
E69C: 20 31 CD	180		JSR	LINGET
E69E: 20 31 C9	181		JSR	FINLIN
E6A2: A5 9B	182		LDA	LOWTR
E6A4: 05 6D	183		STA	DEST
E6A6: A5 9C	184		LDA	LOWTR+1
E6A8: 05 61	185		STA	DEST+1
E6AA: A9 2C	186		LDA	R',
E6AC: 20 E8 01	187		JSR	SYNCHK
E6AE: 20 31 CD	188		JSR	LINGET
	189	*		testar sintaxe
	190	*		Get fim do intervalo (nao
				fazer nada se fim do intervalo
				nao especificado)
E6B2: EA 50	191	INC	LINNUM	Incrementar depois dele, +1


```

E6B4: 00 02 192     BNE N01
E6B6: E6 51 193     INC L1HWR+1
E6B8: 20 31 C9 194     JSR FNDLN      ;Achar linha seguinte
E6BA: A5 9B 195     LDA LOMTR
E6BC: C5 60 196     CMP DEST
E6BE: A5 9C 197     LDA LOMTR+1
E6C0: E5 61 198     SRC DEST+1
E6C2: 80 01 199     BCS MOVOWN    ;Movr, a nao ser que o
E6C4: 60 1100 200     RTS          ;intervalo nao seja valido
E6C6: 40 00 1101
E6C8: 40 00 1102     MOVOWN LDY #000    ;Mov de LOMTR ate VARTAB a DEST
E6CA: B5 9B 1103     MOVIT  LDA (LOMTR),Y
E6CC: 91 60 1104     STA (DEST),Y
E6CE: E6 9B 1105     INC LOMTR
E6D0: 00 02 1106     BNE N02
E6D2: E6 9C 1107     INC LOMTR+1
E6D4: E6 60 1108     INC DEST
E6D6: 00 02 1109     BNE N03
E6D8: E6 61 1110     INC DEST+1
E6DA: A5 A9 1111     LDA VARTAB
E6DC: C5 9B 1112     CMP LOMTR
E6DE: A5 6A 1113     LDA VARTAB+1
E6E0: C5 9C 1114     SRC LOMTR+1
E6E2: 80 E6 1115     BCS MOVIT
E6E4: A6 61 1116     LDX DEST+1    ;Fazer Y,X = DEST-1
E6E6: A4 60 1117     LDY DEST
E6E8: 00 01 1118     BNE NOX
E6EA: CA 1119     DEX
E6EC: 9B 1120     DEY
E6EE: B6 6A 1121     STX VARTAB+1    ;Aponiar VARTAB ao ultimo
E6F0: 84 69 1122     STY VARTAB    byte movido
E6F2: 4C E1 C7 1123     JMP LINKSET    ;Restaurar pointers apos apagar
E6F4: 1124
E6F6: A9 14 1125     GR1  LDA #514
E6F8: 85 22 1126     STA WMDTOP
E6FA: 20 5B FC 1127     JSR NONE
E6FC: 60 1128     RTS
E6FE: 20 5B FC 1129     GR  JSR NONE
E700: 20 F1 E6 1130     JSR GR1
E702: A0 50 C0 1131     LDA INCOLOR
E704: 60 1132     RTS
E706: A0 51 C0 1133     TEXT  LDA 10PER
E708: A9 00 1134     LDA #000
E70A: 80 22 00 1135     STA WMDTOP
E70C: 20 5B FC 1136     JSR NONE
E70E: 60 1137     RTS
E710: 1138
E712: 1139     ;Noticia para guardar e restaurar arrays
E714: 1140     ;(nao usadas para disco)
E716: 1141
E718: 20 5F E3 1142     STONE JSR GETARYPT
E71A: A0 83 1143     LDY #503
E71C: B1 9B 1144     LDA (LOMTR),Y
E71E: AA 1145     TAX
E720: 80 1146     DEY
E722: B1 9B 1147     LDA (LOMTR),Y
E724: E9 01 1148     SRC #501

```

```

E71C: 80 01 149 RCS SLT
E71E: CA 150 DEX
E71F: 85 50 151 SLT STA LINNUM
E721: 86 51 152 STA LINNUM+1
E723: 20 00 FE 153 JSR WRITE
E726: 20 3C E8 154 JSR TAPEPMT
E729: 4C 00 FE 155 JMP WRITE
E72C: 20 59 E8 156 RECALL JSR GETARIFT
E72F: 20 F0 FE 157 JSR READH
E732: A0 02 158 LDT W02
E734: B1 98 159 LDA (LOWTR),Y
E736: C5 50 160 CMP LINNUM
E738: CB 161 INY
E739: B1 98 162 LDA (LOWTR),Y
E73B: E5 51 163 SBC LINNUM+1
E73D: 80 03 164 RCS ROK
E73F: 4C FF C6 165 JMP MONEER
E742: 20 3C E8 166 NOK JSR TAPEPMT
E745: 4C F0 FE 167 JMP READH
340 PUT PARTE20,02
31
32 *Rotinas de inicializacao de alta resolucao
33
E74B: 20 58 FC 14 HGR2 JSR HOME
E74B: A9 17 15 LDA B517
E74D: 85 22 16 STA W0TD0
E74F: 20 58 FC 17 HGR1 JSR HOME
E752: 2C 35 98 18 BIT HGRFLG2
E755: A0 50 C0 19 LDA I0COLOR
E758: 4C 44 E7 110 JMP HTXT
E75B: 20 58 FC 111 HGR JSR HOME
E75E: 20 F1 E6 112 JSR GR1
E761: 2C 54 98 113 BIT HGRFLG1
E764: A0 50 C0 114 HTXT LDA I0COLOR
E767: A5 A8 115 LDA P0AH
E769: 85 E6 116 SETH0D STA HP0D
E76B: A0 57 98 117 LDA SP057
E76E: A0 50 98 118 LDA SP050
E771: A5 00 119 HCLR LDA W000 ;preparar fundo preto
E773: 85 1C 120 STA HCOLOR1
E775: A5 E6 121 BREQD LDA HP0D ;inicializar memoria
E777: 85 18 122 STA SHAPEN alta resolucao
E779: A0 00 123 LDT B500
E77B: 84 1A 124 STY SHAPEN
E77D: A5 1C 125 BREQD1 LDA HCOLOR1
E77F: 91 1A 126 STA (SHAPEN),Y ;fazer alta resolucao=HCOLOR1
E781: 20 FE E7 127 JSR CSMT2 ;Corrigir desvio de cores
E784: CB 128 INY ;Tornar duas vezes mais
E785: 00 F6 129 BNE BREQD1 lento a clear
E787: E6 18 130 INC SHAPEN
E789: A5 18 131 LDA SHAPEN
E78B: 29 1F 132 AND B51F ;Fim?
E78D: 00 E6 133 BNE BREQD1
E78F: 60 134 RTS
135
136 *Rotinas de posicionamento e desenho alta resolucao
137

```

E790:	85	E2	138	HFOSM	STA	Y0		;Entrar com Y em A
E792:	86	E0	139		STX	X0L		; e XL em X
E794:	8A	E1	140		STY	X0H		; e XH em Y
E796:	40		141		PHA			
E79C:	29	D3	142		AND	#5C0		
E79E:	85	D6	143		STA	GBASL		;Para coordenada Y=COORDEN
E798:	4A		144		LSR			;Calcular endereço base
E79C:	4A		145		LSR			;Em GBASL,GBASH para acessar
E79D:	05	D6	146		ORA	GBASL		memoria da tela via (GBASL),Y
E79F:	85	D6	147		STA	GBASL		
E7A1:	68		148		FLA			
E7A2:	85	D7	149		STA	GBASH		
E7A4:	0A		150		ASL			;Calcular
E7A5:	0A		151		ASL			;GBASH+PPFFH0CD
E7A6:	0A		152		ASL			;GBASL+GBASH000
E7A7:	26	D7	153		ROL	GBASH		;com PPP=D01 PARA \$2000 a \$3FFF
E7A9:	0A		154		ASL			se PPP=D10 para \$4000 a \$BFFF
E7AA:	26	D7	155		ROL	GBASH		;Dada coordenada Y=ABCDDEFH
E7AC:	0A		156		ASL			
E7AD:	46	D6	157		ROR	GBASL		
E7AF:	45	D7	158		LDA	GBASH		
E7B1:	29	4F	159		AND	#51F		
E7B3:	09	E6	D0		ORA	HPAG		
E7B6:	85	D7	161		STA	GBASH		
E7B8:	8A		162		TXA			;Dividir X0 por 7 para obter
E7B9:	C0	D0	163		CPT	#500		índice a partir de endereço
E7B9:	F0	D5	164		BCR	#F08D2		base (quociente) e bit dentro
			165	*				do byte de memória da tela
E7BD:	A0	D3	166		LDR	#523		(mascara especific pelo resto)
E7BF:	69	D4	167		AOC	#504		
E7C1:	C8		168	HFOSM	INY			
E7C2:	E9	07	169	HFOSM2	SBC	#6C7		
E7C4:	8D	F8	170		RCS	HFOSM		
E7C6:	84	E5	171		STY	HMSX		;Funciona para X0 de 0 a 279
E7C8:	AA		172		TAX			;Byte de baixa ordem em X
E7C9:	8D	3A	E8	173	LDA	HSHBL-\$F9,X		
E7CC:	85	D0	174		STA	HMSX		;De alta ordem em Y na entrada
E7CE:	98		175		TYA			
E7CF:	4A		176		LSR			
E7D0:	A5	E4	177		LDA	HCOL0R2		;Se byte ímpar (carry ligado)
E7D2:	85	1C	178		STA	HCOL0R1		;Rodar os bits
E7D4:	8D	D8	179		RCS	CSHFT2		
E7D6:	6D		180		RTS			
			181					
E7D7:	2D	9D	E7	182	HPLD1D	JSR	HFOSM	
E7D9:	A5	1C	183		LDA	HCOL0R1		;Calcular pos do bit em EGAS
E7DB:	51	D6	184		ORH	(GBASL),Y		;HMSX e HMSK a partir
E7DE:	25	D0	185		AND	HMSK		das coordenadas Y em A
E7E0:	51	D6	186		ORH	(GBASL),Y		e X em X,Y
E7E2:	91	D6	187		STA	(GBASL),Y		;Para qualquer bit, substituir
E7E4:	A0		188		RTS			bit correspondente de HCOL0R1
			189					
			190					*Rotinas de deslocamento para alta resolucao
			191					
E7E5:	1D	D3	192	LFTNT	BPL	RIGHT		;Usar sinal para rotacao
E7E7:	A5	D0	193	LEFT	LDA	HMSK		esquerda/direita
E7E9:	4A		194		LSR			;Deslocar 7 bits de baixa ordem

E7EA: 80 05	195	BIS	LEFT1	de HMASK de uma posicao
E7EC: 49 C0	196	EOB	HSCO	
E7EE: 85 30	197	LR1	STA	HMASK
E7F0: 60	198		RTS	
	199			
E7F1: 88	1100	LEFT1	DE1	Decrementar indice horizontal
E7F2: 10 02	1101	BPL	LEFT2	
E7F4: A0 27	1102	LDY	#527	Dar a volta na tela
E7F6: A9 C0	1103	LEFT2	LDA	HSCO
E7F8: 85 30	1104	HEXMOX	STA	HMASK, ponto mais a direita do byte
E7FA: 84 E5	1105	STY	HMOX	Atualizar indice horizontal
E7FC: A5 1C	1106	CSHIFT	LDA	HCOLOR1
E7FE: 0A	1107	CSHIFT2	ASL	
E7FF: C9 C0	1108	CMF	HSCO	rodar os 7 bits de baixa ordem de HCOLOR1 de 1 posicao
E801: 10 06	1109	BPL	RTS1	Desviar se branco ou preto
E803: A5 1C	1110		LDA	HCOLOR1
E805: 49 7F	1111	EOB	#57F	
E807: 85 1C	1112		STA	HCOLOR1
E809: 60	1113	RTS1	RTS	
	1114			
E80A: A5 30	1115	RIGHT	LDA	HMASK
E80C: 0A	1116		ASL	
E80E: 49 80	1117	EOB	#580	Deslocar os 7 bits de baixa ordem de HMASK de um bit
E80F: 30 D0	1118	BN1	LR1	
E811: A7 81	1119		LDA	#581
E813: C8	1120	INX		Incrementar byte
E814: 10 28	1121	CPY	#528	
E816: 90 E0	1122	BCC	HEXMOX	
E818: A0 00	1123	LDY	#500	Dar a volta
E81A: 80 D0	1124	BCS	HEXMOX	Sempre
E81C: 18	1125	LRUO1	CLC	Nao ha rotacao de 90 graus
E81E: A5 01	1126	LRUO12	LDA	D1H
E81F: 29 04	1127	AND	#504	Se bit2=0 nao desenhar
E821: F0 25	1128	BEQ	LRU04	
E823: A9 7F	1129	LDA	#57F	
E825: 25 30	1130	AND	HMASK	
E827: 01 26	1131	AND	(GBASL),Y	Ligado bit da tela?
E829: 80 19	1132	BNE	LRU03	
E82B: E6 EA	1133	INC	COLORCOUNT	
E82D: A9 7F	1134	LDA	#57F	
E82F: 25 30	1135	AND	HMASK	
E831: 10 11	1136	BPL	LRU03	Sempre
E833: 18	1137	LRU01	CLC	Nao ha rotacao de 90 graus
E834: A5 01	1138	LRU02	LDA	D1H
E836: 29 04	1139	AND	#504	Se bit2=0 nao desenhar
E838: F0 0E	1140	BEQ	LRU04	
E83A: 81 26	1141	LDA	(GBASL),Y	
E83C: A5 1C	1142	EOB	HCOLOR1	Posicionar bit da tela de alta resolucao com HCOLOR1 corresp
E83E: 25 30	1143	AND	HMASK	
E840: C0 C2	1144	BNE	LRU03	Se mata o bit da tela,
E842: E6 EA	1145	INC	COLORCOUNT	incrementar contador
E844: 51 26	1146	LRU03	EOB	(GBASL),Y
E846: 91 26	1147		STA	(GBASL),Y
E848: A5 01	1148	LRU04	LDA	D1H
E84A: A5 03	1149	ADC	HORINT	Somar esqdrante ao vetor espec
E84C: 29 33	1150	AND	#503	Se mover a esquerda, direita, alto ou baixo conforme
	1151	EOB	=	signal e carry

```

EB4E: 07 02 1152 CMP #102
EB50: 6A 1153 ROR
EB51: 80 92 1154 BCS LTRT
EB53: 30 30 1155 UPDOWN BHI DOWN ;Sinal para escolher alto/baixo
EB55: 18 1156 UPD CLC
EB56: A5 27 1157 LDA GRASH ;Calcular end base para subir
EB58: 2C 36 E9 1158 BIT E01C proxima linha em (GRASH) dando
EB59: 06 22 1159 BNE UP4 a volta em 192
EB5B: 06 26 1160 ASL GRASH
EB5F: 80 1A 1161 BCS UP2
EB61: 2C 40 E9 1162 BIT E03
EB64: F0 0C 1163 BEB UP1
EB66: 6F 1F 1164 ADC #11F ;*** MAPA DE BITS ***
EB68: 38 1165 SEC
EB6F: 80 12 1166 BCS UP3 ;Para linha: ABCDEFGH
EB6B: 6F 23 1167 UP1 ADC #123
EB6D: 48 1168 PHA
EB6E: A5 26 1169 LDA GRASH ;GRASH= E0A0000
EB70: 6F 80 1170 ADC #180 ;GRASH= PFFFF600
EB72: 80 02 1171 BCS UP5
EB74: 6F F0 1172 ADC #FF0 ;Onde FFF= 001 para pag 1
EB76: 85 26 1173 UPS STA GRASH
EB78: 58 1174 PLA
EB7F: 80 02 1175 BCS UP3
EB7B: 6F 1F 1176 UP2 ADC #11F
EB7D: 66 26 1177 UP3 ROR GRASH
EB7F: 6F FC 1178 UP4 ADC #1FC
EB81: 85 27 1179 UPDOWN1 STA GRASH
EB83: A0 1180 RTS
          3A1 PUT PARTEZE_02
          31
EB84: 18 32 CLC
EB85: A5 27 33 DOWN LDA GRASH
EB87: 6F 34 34 ADC #104 ;Calcular end base para descer
          35 E04 " 4-1 proxima linha, dando
EB8F: 2C 36 E9 36 BIT E01C a volta em 192
EB9C: 00 F3 37 BNE UPDOWN1
EB9E: 06 26 38 ASL GRASH
EB9D: 90 19 39 BCC DOWN1
EB9F: 6F E0 310 ADC #FE0
EB9A: 18 311 CLC
EB95: 2C 86 E9 312 BIT E04
EB93: F0 13 313 BEB DOWN2
EB9A: A5 C3 314 LDA GRASH
EB9C: 6F 50 315 ADC #150
EB9E: 6F F0 316 EOR #FF0
EBAD: F0 02 317 BEB DOWN3
EBA2: 6F F0 318 EOR #FF0
EBAA: 85 26 319 DOWN3 STA GRASH
EBAC: A0 E6 06 320 LDAH #PAE
EBAF: 90 02 321 BCC DOWN2
EBAB: 6F E0 322 DOWN1 ADC #1EC
EBAD: 66 26 323 DOWN2 ROR GRASH
EBAF: 96 00 324 BCC UPDOWN1
          325
          326 ;Rotinas para desenhar linhas em alta resolucao
          327

```

EBB1: 40	128	HLINL	FWA	
EBB2: 41 00	129	LDA	WS00	;Fazer XDL, W e YD=0
EBB4: 05 E0	130	STA	XDL	;Para desenhar linha
EBB6: 05 E1	131	STA	XDH	;Relativas a (DX,DY)
EBB8: 05 E2	132	STA	YD	
EBBA: 68	133	PLA		
EBBB: 48	134	HLINL	FWA	;Na entrada ?
EBBC: 38	135	SEC		;XL: A
EBBD: E5 E0	136	SBC	XDL	;XH: X
EBBF: 48	137	FWA		;Y: Y
EBD0: 0A	138	TAA		
EBD1: E5 E1	139	SBC	XDH	
EBD3: 95 D3	140	STA	GORNT	;Calcular ABS (X-YD) em (DX,H)
EBD5: 80 DA	141	RCS	HLIN2	
EBD7: 68	142	PLA		
EBD8: 49 FF	143	EOR	WSFF	;X para o bit de sinal do quadr
EBDA: 49 D1	144	ADC	WS01	
EBDC: 48	145	FWA		;D= direita (DX positivo)
EBDD: 49 D0	146	LDA	WS00	;L= esquerda (DX negativo)
EBDF: E5 D3	147	SBC	GORNT	
EBE1: 85 D1	148	HLIN2	STA	DH
EBE3: 05 D5	149		STA	EH
EBE5: 68	150	PLA		;Inicializar EL,H com
EBE6: 85 D0	151	STA	DXL	;ABS (X-YD)+ DELTX
EBE8: 85 D4	152	STA	EL	
EBEA: 68	153	PLA		
EBEB: 05 E0	154	STA	XDL	;Coloc pointer no fim da linha
EBED: 86 E1	155	STX	EH	
EBEF: 98	156	TYA		
EBF0: 18	157	CLC		
EBF1: E5 E2	158	SBC	YD	;Calcular -DELTY-L em D1 onde
EBF3: 90 D4	159	ECC	HLIN3	;DELTY = ABS (Y-YD)
EBF5: 49 FF	160	EOR	WSFF	
EBF7: 49 FE	161	ADC	WSFE	
EBF9: 85 D2	162	HLIN3	STA	DY
EBFB: 84 E2	163		STY	YD
EBFD: 66 D3	164	ROR	GORNT	;ID= alto, 0=baixo
EBFF: 38	165	SEC		
EBF0: E5 D0	166	SBC	DXL	;Inicializ COUNTL,H (COUNTL= 1)
EBF2: A8	167	TAX		;com -(DELTX + DELTY + 1)
EBF3: 49 FF	168	LDA	WSFF	
EBF5: E5 D1	169	SBC	DXH	
EBF7: 85 D3	170	STA	COUNTH	
EBF9: A4 E5	171	LDT	HMOX	;Indice horizontal
EBFB: 80 D5	172	RCS	MOVEX2	;Sewer
EBFD: 0A	173	MOVEX	ASL	;Mover na direcao X
EBFE: 2D E5 E7	174	JSR	LFTRT	;Usar bit 6 de GORNT para
EY01: 38	175	SEC		;posicionar esquerda/direita
EY02: A5 D4	176	MOVEX2	LDA	EL
EY04: A5 D2	177		ADC	DY
EY06: 85 D4	178	STA	EL	;Fazer (EL,H) = DELTY = (EL,H)
EY08: A5 D5	179	LDA	EH	;Nota: DY e (DELTY) - 1
EY0A: E9 D0	180	SBC	WS00	;Desl carry se (EL,H) fica NEG.
EY0C: 85 D5	181	HLIN2	STA	EH
EY0E: 81 26	182	LDA	(GBASL),Y	
EY10: A5 1C	183	END	HCOLOR1	;Desenhar posto de HCOLOR1
EY12: 25 3C	184	AND	HBACK	

E914:	51 26	185	EM	(GBASL),Y	
E916:	91 26	186	STA	(GBASL),Y	
E918:	EB	187	INX		;Feitos os pto (DELTX+DEITY) ?
E919:	00 04	188	RMC	HLIN4	
E91B:	E6 10	189	INC	COUNTH	
E91D:	F0 62	190	SEN	R152	;Sim, voltar
E91F:	A5 03	191	LDA	008MT	;Para testar direcao
E921:	80 0A	192	RCS	MOVEX	;Se carry lig, desenhar (EL,H)
E923:	20 53 E8	193	JSR	UP06AH	;Se desligado negativo, mover
E926:	10	194	CLC		
E927:	A5 04	195	LDA	EL	; (EL,H) + DELTX EM (EL,H)
E929:	A5 00	196	ADC	00L	
E92B:	85 04	197	STA	EL	
E92D:	A5 05	198	LDA	EH	
E92F:	A5 01	199	ADC	00H	
E931:	50 09	1100	RVC	HCAUMT	;Sempre
		1101			
E933:	81	1102	MSKTEL	DFB	Z10000001
E934:	82	1103		DFB	Z10000010
E935:	84	1104		DFB	Z10000100
E936:	86	1105		DFB	Z10001000
E937:	90	1106		DFB	Z10010000
E938:	A0	1107		DFB	Z10100000
E939:	C0	1108		DFB	Z11000000
		1109			
E93A:	1C	1110	EMIC	HEX	1C
		1111			
		1112			*Tabela de COS (90°/16graus)*5100 - 1
		1113			com um byte de precisao, X de 0 a 16
		1114			
E93B:	FF FE FA	1115	COSTEL	HEX	FFFEF4ECE104C5
E93E:	F4 EC E1 04 C5	1116		HEX	846180781493158
E943:	84 A1 80	1117		HEX	FF
E946:	78 61 49	1118			
E94B:	FF	1119			*Rotina que resfzara coordenadas de alta resolucao
		1120			
E94C:	A5 26	1121	HFIMO	LDA	GBASL
E94E:	0A	1122		ASL	;Converter endereco base
E94F:	A5 27	1123	LDA	GBASH	;A coordenada T
E951:	29 03	1124	AND	#503	
E953:	2A	1125	RCL		;Para fazer (GBASL= ENAB000
E954:	05 26	1126	ORA	GBASL	GBASH= PPPF6HCO
E956:	0A	1127	ASL		
E957:	0A	1128	ASL		
E958:	0A	1129	ASL		;Coordenada T= ABCDEF6H
E959:	85 E2	1130	STA	Y0	
E95B:	A5 27	1131	LDA	GBASH	;PPP= pagina de tela
E95D:	4A	1132	LSR		
E95E:	4A	1133	LSR		
E95F:	29 07	1134	AND	#507	
E961:	05 E2	1135	ORA	Y0	
E963:	85 E2	1136	STA	Y0	;Converter HMOX (indice do
E965:	A5 E5	1137	LDA	HMOX	endereco base) e HMOX
E967:	0A	1138	ASL		;mascara de bit) a coord
E968:	A5 E5	1139	ADC	HMOX	X em (X0L,H), intervalo 0-5133

E96A:	0A	1140	ASL	
E96B:	AA	1141	TAX	
E96C:	CA	1142	DEX	
E96D:	AS 30	1143	LDA HMASK	
E96F:	29 7F	1144	AND #57F	
E971:	EB	1145	HFIND1	DIX
E972:	4A	1146	LSR	
E973:	00 FC	1147	ONE HFIND1	
E975:	85 E1	1148	STA XDM	
E977:	8A	1149	TXA	
E978:	18	1150	CLC	;Calcular HMDX*7 + LOG(HMASK), base 2
E979:	45 E5	1151	ADC HMDX	
E97B:	90 02	1152	BCC HFIND2	
E97D:	E4 E1	1153	INC XDM	
E97F:	85 E0	1154	HFIND2 STA XDL	
E981:	60	1155	RTS2	RTS
		1156		
		1157	;Rotinas para desenhar formas em alta resolucao	
		1158		
E982:	B6 1A	1159	DRAW STX SHMPEL	;Pointer a forma
E984:	84 18	1160	STY SHMPEN	
E986:	AA	1161	DRAW1 TAX	;A conta ROT (0 - 53F)
E987:	4A	1162	LSR	
E988:	4A	1163	LSR	
E989:	4A	1164	LSR	
E98A:	4A	1165	LSR	
E98B:	85 D3	1166	STA GORNT	;GORNT 0= alto, 1= direita
E98D:	BA	1167	TXA	;2=baixo, 3=esquerda
E98E:	29 0F	1168	AND #50F	
E990:	AA	1169	TAX	
E991:	BC 38 E9	1170	LDY COSTBL,X	;Guardar COS e SEN em BL e DY
E994:	84 00	1171	STY DXL	
E996:	49 0F	1172	EDR #50F	
E998:	AA	1173	TAX	
E999:	BC 3C E9	1174	LDY COSTBL+1,X	
E99C:	C0	1175	IMY	
E99D:	84 02	1176	STY DY	
E99F:	A4 E5	1177	LDY HMDX	;Indice do endereco base
E9A1:	A2 00	1178	LSR #500	
E9A3:	86 EA	1179	STX COLCOUNT	;Limpar contador de colisoes
E9A5:	A1 1A	1180	LDA (SHMPEL,X)	;Primeiro byte da def de formas
E9A7:	85 D1	1181	STA DDM2	
E9A9:	A2 80	1182	LDR #580	
E9AB:	86 04	1183	STX EL	;EL,W para vetores alto, baixo,
E9AC:	86 05	1184	STX EH	esquerda, direita fracionarios
E9AF:	A6 E7	1185	LDR SCALEZ	;Fator de escala
E9B1:	A5 04	1186	LDW EL	
E9B3:	38	1187	SEC	;Se cosseco fracionario da
E9B4:	45 00	1188	ADC DXL	overflow, mover na direcao
E9B6:	85 04	1189	STA EL	do vetor especificado
E9B8:	90 04	1190	BCC DRAW4	
E9BA:	20 33 EB	1191	JSR LRU01	
E9BD:	18	1192	CLC	
E9BE:	A5 05	1193	LDW EH	;Se a seno fracionario da
E9C0:	45 02	1194	ADC DY	overflow mover na direcao
E9C2:	85 05	1195	STA EH	especificada + 90 graus
E9C4:	90 03	1196	BCC DRAW5	

EYCA: 20 34 EB	1197	JSR	LR00C	
ERYA: CA	1198	DRAW5	SEX	;Loop cont o fator de escala
EYCA: 00 E5	1199		BNE	DRAW3
EYCC: A5 D1	1200		LDA	DXM
EYCE: 4A	1201		LSR	
EYCF: 4A	1202		LSR	;Proximos 3 bits do vetor de definicao de forma
EYD0: 4A	1203		LSR	
EYD1: 00 D4	1204		BNE	DRAW2
EYD3: E6 1A	1205		INC	SHAPEL
EYD5: 00 D2	1206		BNE	MT4
EYD7: E6 1B	1207		INC	SHAPEL+1
EYD9: A1 1A	1208	MT4	LDA	(SHAPEL,X)
ERD0: 00 CA	1209		BNE	DRAW2
ERD1: 60	1210		RTS	;Se 0, fim
	1242		PUT	PARTZF,02
	12			
	13			Rotina para desenhar formas (XDRAW) em alta resolucao
	13			
ERYE: 06 1A	14	XDRAW0	STX	SHAPEL
ERYD: 04 1B	15		STY	SHAPEM
ERYE: 0A	16	XDRAW0	TAX	;A contem RTT (0 - \$3F)
ERYC: 4A	17		LSR	
ERYE: 4A	18		LSR	
ERYE: 4A	19		LSR	
ERYE: 4A	110		LSR	
ERYE: 05 D3	111		STA	GORNT
ERYE: 0A	112		TBA	
ERYE: 2F DF	113		AND	MSOF
ERYE: 0A	114		TAX	
ERYD: 0C 3B E9	115		LDY	COSTBL,X
ERYD: 04 D0	116		STY	D1L
ERYD: 4F DF	117		FOR	MSOF
ERYA: 0A	118		TAX	
ERYD: 0C 3C E9	119		LDY	COSTBL+1,X
ERYB: C8	120		INY	
ERYF: 04 D2	121		STY	DY
ERYB: 0A E5	122		LDY	MBIX
ERYD: A2 D0	123		LDX	MSOD
ERYF: 06 E6	124		STX	COLCOUNT
EAD1: A1 1A	125		LDA	(SHAPEL,X)
EAD3: 05 D1	126	XDRAW2	STA	DOH
EAD5: A2 D0	127		LDX	MSBD
EAD7: 06 D4	128		STX	EL
EAD7: 06 D5	129		STX	EH
EAD8: A6 E7	130		LDX	SCALEZ
EAD0: 45 D4	131	XDRAW3	LDA	EL
EADF: 3B	132		SEC	
EAD0: 45 D0	133		ADC	D1L
EAD2: 05 D4	134		STA	EL
EAD4: 9D D4	135		BCC	XDRAW4
EAD4: 2D 1C E8	136		JSR	LRUDX1
EAD9: 1B	137		CLC	
EAD1: 45 D5	138	XDRAW4	LDA	EH
EAD1: 45 D2	139		ADC	D1
EAD1: 05 D5	140		STA	EH
EAD0: 9D D3	141		BCC	XDRAW5
EAD2: 2D 1B E8	142		JSR	LRUDX2

```

EA25: 0A      >43  XDRW5  DEI      ;loop cont o fator de escala
EA26: 00 E5   >44      BNE  XDRW3
EA2B: A5 D1   >45      LDA  D0H
EA2A: 4A      >46      LSR      ;Proximos 3 bits do vetor de
EA2B: 4A      >47      LSR      ; definicao de forma
EA2C: 4A      >48      LSR
EA2D: 00 04   >49      BNE  XDRW2   ;falta este byte
EA2F: E6 5A   >50      INC  SHAPEL
EA31: 00 02   >51      BNE  N15
EA33: E6 5B   >52      INC  SHAPEL+1
EA35: A5 1A   >53  N15      LDA  (SHAPEL,X) ;prox. byte da def. de formas
EA37: 03 CA   >54      BNE  XDRW2   ; se 0, fim
EA39: 60      >55      RTS
                >56
                >57  ;Rotinas de analise usadas pelo BASIC para
                >58  ;acesso em alta resolucao
                >59
EA3A: 20 BF D0 >60  HFW5     JSR  FRMMH
EA3D: 20 6C D4 >61          JSR  GETADR
EA40: A4 51   >62          LDY  LINNUM+1 ;Obter coord horizontal em X,Y
EA42: A6 5D   >63          LDX  LINNUM
EA44: C0 01   >64          CPY  #1280 ;Testar se i < 280
EA46: 9D 06   >65          BCC  HFW51
EA4B: 00 0D   >66          BNE  GEERR
EA4A: ED 18   >67          CPX  #1280
EA4C: B0 19   >68          BCS  GEERR
EA4E: BA      >69  HFW51     TBA      ;Guardar coord horiz na pilha
EA4F: 4B      >70          PHA
EA50: 9B      >71          TBA
EA51: 4B      >72          PHA
EA52: A9 2C   >73          LDA  #' '
EA54: 20 EB D1 >74          JSR  SYMCHR ;Testar sintaxe
EA57: 20 12 D4 >75          JSR  GETBYT ;Obter coordenada vertical
EA5A: E0 C0   >76          CPX  #192 ;Testar se i < 192
EA5C: B0 0F   >77          BCS  GEERR
EA5E: 86 9D   >78          STX  DSCTHP ;Guardar
EA60: 68      >79          PLA      ;Recuperar coord horizontal
EA61: AB      >80          TAY
EA62: 68      >81          PLA
EA63: AA      >82          TAX
EA64: A5 9D   >83          LDA  DSCTHP ;E coordenada vertical
EA66: 60      >84          RTS      ;Voltar ao prog que chamou
                >85
EA67: 4C 5A E5 >86  GEERR     JMP  GEERR ;quantidade valida
                >87
EA6A: C9 D0   >88  HCOLOR    CMP  #100
EA6C: 03 03   >89          BNE  EVVAR
EA6E: 20 B1 D3 >90          JSR  CHGET
                >91
EA71: 20 12 D4 >92  EVVAR     JSR  GETBYT
EA74: E0 0B   >93          CPX  #100
EA76: B0 EF   >94          BCS  GEERR
EA7B: B0 AA FB >95          LDA  TABCOR,X
EA7B: B5 E4   >96          STA  HCOLORZ
EA7D: 60      >97          RTS
                >98
EA7E: C9 C1   >99  HPL31     CMP  #0 ;Pedido desenho continuo?

```

EAB0: F0 00	:100	BEQ	HF3	;fim, desviar	
EAB2: 20 3A EA	:101	JSR	HFMS	;Obter coord do pto de partida	
EAB5: 20 07 E7	:102	JSR	HPL0T0	;Desenha-la, preparando coord	
EAB8: 20 07 00	:103	JSR	CHRG0T		
EAB9: C9 C1	:104	CMP	#to	;linha especificada?	
EABD: 00 0E	:105	RNE	RTS3	;Nao, sair	
EABF: 20 E8 06	:106	JSR	SYNCH		
EAP2: 20 3A EA	:107	JSR	HFMS	;Obter coord do fim da linha	
EAP5: 84 90	:108	STY	DSCTMP	;Preparar linha	
EAP7: A0	:109	TAY			
EAP8: 8A	:110	TXA			
EAP9: A6 90	:111	LDX	DSCTMP		
EAPB: 20 88 E8	:112	JSR	HELTN	;Desenhar linha	
EAPF: 4C 88 EA	:113	JMP	HP2	;Loop ate nao ter mais "00"	
	:114				
EAA1: 20 12 0A	:115	ROT	JSR	GETBYT	
EAA4: 86 F9	:116	STX	ROTZ		
EAA6: 60	:117	RTS			
	:118				
EAA7: 20 12 0A	:119	SCALE	JSR	GETBYT	
EAA7: 86 E7	:120	STX	SCALEZ		
EAAE: 60	:121	RTS			
	:122				
EAA0: 20 12 0A	:123	SHAPEPNT	JSR	GETBYT	;Numero de forma especificado
EAB0: A5 E8	:124	LDA	SHAPEPNT		
EAB2: 85 1A	:125	STA	SHAPEL		
EAB4: A5 E9	:126	LDA	SHAPEPNT+1		
EAB6: 85 1B	:127	STA	SHAPEH		
EAB8: 8A	:128	TBA			
EAB9: A2 00	:129	LDX	#SD0		
EABE: C1 1A	:130	CMP	(SHAPEL,X)	;Forma definida na tabela?	
EABD: F0 02	:131	BEQ	DP1		
EABF: 80 A6	:132	BCC	GGERR	;Nao, erro	
EAC1: 0A	:133	ASL		;Achar end da forma na tabela	
EAC2: 90 03	:134	BCC	DP2		
EAC4: E6 1B	:135	INC	SHAPEH		
EAC6: 1B	:136	CLC			
EAC7: A8	:137	DP2	TAY		
EAC8: B1 1A	:138	LDA	(SHAPEL),Y		
EACA: A5 1A	:139	ADC	SHAPEL		
EACC: 0A	:140	TAX			
EACD: C8	:141	INY			
EACE: B1 1A	:142	LDA	(SHAPEL),Y		
EAD0: A5 E9	:143	ADC	SHAPEPNT+1		
EAD2: 85 1B	:144	STA	SHAPEH	;Guardar endereco da forma	
EAD4: 86 1A	:145	STX	SHAPEL		
EAD6: 20 07 00	:146	JSR	CHRG0T	;Testar sintaxe	
EAD9: C9 C3	:147	CMP	#at		
EAD6: D9 09	:148	RNE	DP3		
EAD0: 20 E8 01	:149	JSR	SYNCH		
EAD0: 20 3A EA	:150	JSR	HFMS	;Onde desenhar a forma?	
EAE3: 20 90 E7	:151	JSR	HPOSH		
EAE6: A5 F9	:152	LDA	ROTZ		
EAE8: 60	:153	RTS			
	:154				
EAE9: 20 A0 EA	:155	DRND	JSR	DRNPNT	;Entradas do BASIC
EAE3: 4C 86 E7	:156	JMP	DRND1		

```

EAF7: 20 AD EA 1157 XORAW JSR DRWPNT
EAF8: 4C E2 E9 1158 JMP XORAW
1159
1160 *Carregar tabela de formas a partir de fita
1161
EAF5: AF 00 1162 SHLWD LDA W500
EAF7: 05 30 1163 STA A1H
EAF9: 05 3F 1164 STA A2H
EAFB: AD 50 1165 LDY BLINUM
EAFD: 04 3C 1166 STY A1L
EAF7: 08 1167 INC
EBD0: 84 3E 1168 STY A2L
EBD2: 20 FD FE 1169 JSR READR ;ler tamanho em LINNUM
EBD5: 18 1170 CLC
EBD6: A5 73 1171 LDA NENSIZ
EBD8: A6 1172 TAX
EBDA: C6 1173 DEX
EBDC: 86 3E 1174 STX A2L ;NENSIZ-1 -> pointer ao fim
EBDE: E5 50 1175 SRC LINUM ;NENSIZ-2 tamanho -> pointer
EBDF: 48 1176 PHA ;ao começo
EBE0: A5 74 1177 LDA NENSIZ+1
EBE1: A8 1178 TAX
EBE2: E8 1179 INC
EBE3: 00 01 1180 BNE SL1
EBE5: 88 1181 DEY
EBE6: 04 3F 1182 SL1 STY A2H
EBE8: E5 51 1183 SRC LINUM+1
EBEA: C5 6E 1184 CMP STREND+1
EBEC: 90 02 1185 BCC SL2
EBEE: 00 00 1186 BNE SL3 ;pagina deve ser > STREND
EBF0: 4C FF C6 1187 SL2 JMP MEMOR
EBF2: 85 74 1188 SL3 STA NENSIZ+1 ;Por em HONEN e pointer as
EBF5: 85 70 1189 STA FRETOP+1 ;formas valor da tabela de
EBF7: 05 30 1190 STA A1H ;formas
EBF9: 85 E9 1191 STA SHAPFNT+1
EBFB: 68 1192 PLA
EBFD: 85 E8 1193 STA SHAPFNT
EBFE: 85 73 1194 STA NENSIZ
EB00: 85 AF 1195 STA FRETOP
EB02: 85 3C 1196 STA A1L
EB04: 20 FA FC 1197 JSR BGRIT ;levar aresta
EB07: A9 03 1198 LDA W503
EB09: 4C 02 FF 1199 JMP READ2 ;ler tabela de formas em (A1L)
119A PUT PARTE2,02
119B
EB0C: 18 119C TAPEFNT CLC
EB0E: A5 98 119D LDA LONTR
EB10: A5 50 119E ADC LINUM
EB12: 85 3E 119F STA A2L
EB14: A5 9C 119D LDA LONTR+1
EB16: A5 58 119E ADC LINUM+1
EB18: 85 3F 119F STA A2H
EB1A: A0 04 119D LDY W504
EB1C: 81 98 119D LDA (LONTR),Y
EB1E: 20 D9 04 119D JSR GETMY2
EB20: A5 94 119E LDA H5B03
EB22: 85 3C 119F STA A1L

```

```

EB54: A5 95 114      LDA HIGHOS+1
EB56: B5 30 115      STA A5H
EB58: 60 116      RTS
117
118      ;Notina nao grafica
119
EB59: A9 40 120      GETARYPT LDA #540      ;Chamada por STORE e RECALL
EB5B: B5 14 121      STA SUBFLG
EB5D: 20 03 03 122      JSR PTWGET
EB60: A9 00 123      LDA #500
EB62: B5 14 124      STA SUBFLG
EB64: AC 15 CC 125      JMP VWR710
126
127      ;Observar que se WNOLEFT nao for 0, HTAB pode
128      ;imprimir fora da tela, isto e, no programa
129
EB67: 20 12 DA 130      HTAB JSR GETBYT
EB6A: CA 131      DEC
EB6B: BA 132      TRA
EB6C: C9 28 133      HTAB1 CMP #128
EB6E: F0 DA 134      BCC HTAB2
EB70: E9 28 135      SBC #128
EB72: 40 136      PHA
EB73: 20 20 CE 137      JSR CRD0
EB76: A0 138      PLA
EB77: AC 6C EB 139      JMP HTAB1
EB7A: B5 24 140      HTAB2 STA CH
EB7C: 60 141      RTS
142
EB7D: 20 12 DA 143      MOTOR JSR GETBYT
EB80: BA 144      TRA
EB81: C9 04 145      CMP #5DA      ;E motor ?
EB83: F0 02 146      BCC A0B8?      ;Sim
EB85: B0 68 147      BCS JSCHVIR
EB87: C9 02 148      A0B8? CMP #5B2      ;Motor A ou B ?
EB89: F0 02 149      BCC M0TRA
EB8B: 69 01 150      ADC #5B1      ;Se motor B soma 5ccarr
EB8D: AA 151      MOTRA TAB
EB8E: B0 5C 03 152      LDA MOTABFF,X      ;L gal/desi A ou B
EB91: 60 153      RTS
154
EB92: 20 17 00 155      NL/NO JSR CRIGET
EB95: C9 41 156      CMP #541      ;Compara primeira letra com A
EB97: D0 05 157      BNE CRIGFLT
EB99: EA 86 158      INC 11TPTR
EB9B: AC FE CB 159      JMP LOW
EB9E: 20 6E EB 160      CRIGFLT JSR CRKT      ;Check se a proxima letra e T
EBA1: AA 8B 161      LDY 11TPTR
EBA3: 20 6F EC 162      JSR K0200
EBA6: BA 86 163      STY 11TPTR
EBAB: 20 AE ED 164      JSR LOADT
EBAD: A0 6E 04 165      LDA PGR1
EBAE: AC 6F 04 166      LDY PGR1+1
EBB1: B5 67 167      STA BASLOC      ;Indica o inicio do programa
EBB3: BA 68 168      STY BASLOC+1
EBB5: A0 6A 04 169      LDA PORC
EBB8: AC 6B 04 170      LDY PORC+1

```

EBB8: 85 69	171		STA	WRTAB	!Indica o fim do programa
EBB8: 84 6A	172		STY	WRTAB+1	
EBBF: 85 AF	173		STA	BASLOC1-1	
EBF1: 84 8D	174		STY	BASLOC1	
EBF3: 6D	175		RTS		
	176				
EBF4: 2D 87 08	177	MSAVE	JSR	CHKDOT	
EBF7: C9 41	178		CMF	B541	!Compara proxima letra com A
EBF9: 0D 05	179		BNE	CHKDFST	
EBFB: E6 88	180		JMC	TXTPTR	
EBFD: 4C 85 C8	181		JMP	SAVE	!SAVE no formato APPLE
EBFD: 2D EE EB	182	CHKDFST	JSR	CHKT	!Compara se a proxima letra e T
EBF3: A5 67	183		LDA	BASLOC	
EBF5: A4 68	184		LDY	BASLOC+1	
EBF7: 85 3C	185		STA	A1L	
EBF9: 84 3D	186		STY	A1H	
EBFB: A5 AF	187		LDA	BASLOC1-1	
EBFD: A4 8D	188		LDY	BASLOC1	
EBFD: 85 3E	189		STA	A2L	
EBF1: 84 3F	190		STY	A2H	
EBF3: A4 88	191		LDY	TXTPTR	
EBF5: 2D EF EE	192		JSR	KEYWD	
EBF8: 84 88	193		STY	TXTPTR	
EBFA: 2D 9A EC	194		JSR	SAVET	
EBFD: 6D	195		RTS		
	196				
EBFE: C9 54	197	CHKT	CMF	B541	!Compara com T
EBFD: FD 03	198		BEQ	ADVIND	
EBF2: 4C F1 01	199	JSYNERR	JMP	SYNERR	
EBF5: E6 88	1100	ADVIND	JMC	TXTPTR	!Avanca keyword input buffer
EBF7: 6D	1101		RTS		
	1102				
EBF8: AF 4D	1103	MP	LDA	B54D	!End alto da segunda pag. de video
EBFA: 85 48	1104		STA	PDWH	
EBFC: AD 35 C0	1105		LDA	DISPAG2	!Chaveia segunda pag. de video
EBFF: 6D	1106		RTS		
	1107				
EC00: A9 2D	1108	MA	LDA	MS2D	!End alto da prima. pag. de video
EC02: 85 48	1109		STA	PDWH	
EC04: AD 54 C0	1110		LDA	DISPAG1	!Chaveia primeira pag. de video
EC07: 6D	1111		RTS		
	1112				
EC08: FF FF	1113		HEX	FFFF	
EC0A: 85 F8	1114	CONT3	STA	COOL	!Salva caracter de saida
EC0C: A5 47	1115		LDA	POWL	
EC0E: F8 05	1116		BEQ	ISV0	!POWL=0 ?
EC10: A5 F8	1117		LDA	COOL	
EC12: 4C 8F 08	1118		JMP	PR09	
	1119				
EC15: AD 52 04	1120	ISV0	LDA	GRAFFLG	!Ocorre ou BF2 antes?
EC18: 0D 2D	1121		BNE	CHGRF	!Sim,entao e caracter grafico
EC1A: A5 49	1122		LDA	VCTR	!Ocorre ou CTRL-B antes?
EC1C: 0D 42	1123		BNE	ISCTR?	!Sim entao va p/ ISCTR?
EC1E: A5 F8	1124		LDA	COOL	!Pegar caractere
EC20: 3D 3E	1125		BEQ	ISCTR?	!Se ASCII
EC22: 2D EA F7	1126		JSR	SELEFAX	!Seleciona graf.ou palavras BASIC
EC25: 8D 05	1127		BES	ISGRF?	!Se graficos

EC27:	A5 F8	>120	LDA	COOL	;Pega caracter
EC29:	4C 49 F7	>129	JMP	ISCRAS	;Imprime comandos BASIC
		>130			
EC2C:	A5 49	>131	ISGRF?	LDA	VCTR
EC2E:	D0 18	>132	BNE	ISRET?	
EC30:	A5 F8	>133	LDA	COOL	
EC32:	C9 34	>134	"NP	#534	;Se caracter !=34,entao ISRET?
EC34:	80 12	>135	BOS	ISRET?	
EC36:	CA	>136	DEX		
EC37:	4C 3A FF	>137	JMP	BELL	;Se esta entre 1E e 34 gera um BELL
		>138			
EC3A:	A5 F8	>139	CHGRF	LDA	COOL
EC3C:	29 7F	>140	AND	#57F	;Transforma em positivo ASCII
EC3E:	38	>141	SEC		
EC3F:	E9 40	>142	SBC	#540	;Caracter graficos
EC41:	85 F8	>143	STA	CHOL	
EC43:	A9 00	>144	LDA	#500	
EC45:	80 52 04	>145	STA	GRAFFLG	
EC4B:	A5 F8	>146	ISRET?	LDA	COOL
EC4A:	C9 00	>147	CMP	#580	;E RETURN?
EC4C:	00 00	>148	BNE	NTRET	
EC4E:	A0 82 04	>149	LDA	FIMLIN	
EC51:	F0 06	>150	BEQ	NTRET	
EC53:	20 62 FC	>151	JSR	CR	
EC56:	A9 00	>152	LDA	#500	
EC58:	80 82 04	>153	STA	FIMLIN	
EC5B:	A5 F8	>154	NTRET	LDA	COOL
EC5D:	4C A9 F1	>155	JMP	ISCTP	;Saida de caracter
EC60:	A5 F8	>156	ISCTB?	LDA	COOL
EC62:	C9 80	>157	CMP	#582	;Compara com CTRL-B
EC64:	F0 0A	>158	BEQ	ISCTB	
EC6A:	C9 F2	>159	CMP	#graf	
EC6B:	00 C2	>160	BNE	ISGRF?	
EC6A:	A9 01	>161	LDA	#501	
EC6C:	80 52 04	>162	STA	GRAFFLG	
EC6F:	60	>163	RTS		
		>164			
EC70:	A9 01	>165	ISCTB	LDA	#501
EC72:	45 49	>166	EOR	VCTR	;Inverte VCTR
EC74:	85 49	>167	STA	VCTR	
EC76:	4C 2C EC	>168	JMP	ISGRF?	
		>169			
EC79:	A2 FF	>170	PRPRE0	LDX	#5FF
EC7B:	8E 88 C0	>171	STX	PRF2	
EC7E:	E8	>172	TXI		;Zera 446, POUT, VCTR
EC7F:	8A	>173	TXA		
EC80:	85 46	>174	STA	446	
EC82:	85 47	>175	STA	POUL	
EC84:	85 49	>176	STA	VCTR	
EC86:	20 0C F1	>177	JSR	ZPE4	;Zera pagina 4
EC89:	60	>178	RTS		
		>179			
EC8A:	A0 41 04	>180	TESFLG	LDA	TEST41
EC8D:	F0 C0	>181	BEQ	SET442	
EC8F:	4C 0A F1	>182	JMP	CONTCR	
EC92:	A0 01	>183	SET442	LDY	#501
EC94:	8C 42 04	>184	STY	TEST42	

EC97: 4C C2 F1 1185	JMP	HTCIP	
1186			
EC9A: 2D 9F ED 1187	SAVET	JSR	TFM
EC9B: 28 1188		SEC	
EC9E: A5 3F 1189		LDA	A2H
ECAD: E5 30 1190		SBC	ALH
ECAC: 80 46 04 1191		STA	BUF+6
ECAS: A5 3E 1192		LDA	ACL
ECAT: E5 3C 1193		SBC	ACL
ECAR: 90 03 1194		CC	IFBORN
ECAB: EE 46 04 1195		INC	BUF+6
ECAE: A2 00 1196	IFBORN	LDX	#500
ECB0: BE 50 04 1197		STX	TEST50
ECB3: CA 1198		DEX	
ECB4: BE 67 04 1199		STX	BUF+7
ECB7: BE 51 04 1200		STX	TEST51
ECBA: A9 24 1201		LDA	#524
ECBC: A0 2D 1202	WXZER00	LDY	#520
ECBE: 2D 08 FC 1203		JSR	ZERDLY
ECD1: 00 F9 1204		RME	WXZER00
ECDB: 69 FE 1205		ADC	#5FE
ECDE: 80 F5 1206		BCS	WXZER00
ECDF: 2D 19 ED 1207	MOBSAV	JSR	PRP
ECEA: A9 40 1208		LDA	#540
ECEC: A0 50 1209	WXZERD1	LDY	#550
ECEE: 2D 06 FC 1210		JSR	ZERDLY
ECD1: 69 FE 1211		ADC	#5FE
ECD3: 80 F7 1212		BCS	WXZERD1
ECD5: A0 21 1213		LDY	#521
ECD7: 2D 06 FC 1214		JSR	MRB1T
ECD4: EE 67 04 1215		INC	BUF+7
ECD0: A0 67 04 1216		LDA	BUF+7
ECE0: 80 53 04 1217		STA	TEST53
ECE3: A9 FF 1218		LDA	#5FF
ECE5: A0 22 1219		LDY	#522
ECE7: 2D 2A ED 1220		JSR	WFGM
ECEA: 40 1221	PMA		
ECEB: 2D 30 ED 1222		JSR	CSST1
ECEE: 48 1223		PLA	
ECEF: 2D 2A ED 1224		JSR	WPSM
ECF2: A0 21 1225		LDY	#521
ECF4: 2D ED FE 1226		JSR	WRYTE
ECF7: A0 67 04 1227		LDA	BUF+7
ECFA: F0 CB 1228		BEQ	MOBSAV
ECFC: A5 3C 1229		LDA	AIL
ECFE: 80 7D 04 1230		STA	TEST70
ED01: A5 30 1231		LDA	ALH
ED03: 8D 71 04 1232		STA	TEST71
ED05: A0 50 04 1233		LDA	TEST50
ED0F: 00 02 1234		RME	FINSAVE
ED0B: F0 BA 1235		BEQ	MOBSAV
ED0F: A2 03 1236	FINSAVE	LDX	#503
ED0F: A0 00 1237		LDY	#500
ED11: 2D F3 FA 1238		JSR	NEWS
ED14: A9 87 1239		LDA	#587
ED16: 4C 09 FB 1240		JMP	BELLO
1241			

;Se borrow

;Escreve um bit

;Inicializa com #5FF

;Salva CHKSUM

;Recupera CHKSUM

;Escreve

;Escreve um CHKSUM

;Imprime a mensagem "OK"


```

ED17: A9 6D 1242 FRP LDA #560
ED18: 85 3C 1243 STA A1L
ED1D: A9 04 1244 LDA #504
ED1F: 85 3D 1245 STA A1H
ED21: A9 67 1246 LDA #567
ED23: 85 3E 1247 STA A2L
ED25: A9 04 1248 LDA #504
ED27: 85 3F 1249 STA A2H
ED29: 60 1250 RTS
      1251
ED2A: A2 00 1252 MPRM LDY #500
ED2C: 41 3C 1253 EOR (A1L,X) ;CHRSUM
ED2E: 40 1254 PHA
ED2F: A1 3C 1255 LDA (A1L,X)
ED31: 20 ED FE 1256 JSR WRYTE ;Escrive um byte
ED34: 20 BA FE 1257 JSR NEXTAL ;Proximo Al ate A1=A2
ED37: AD 1D 1258 LDY #510
ED39: 68 1259 PLA
ED3A: 70 EE 1260 ROC MPRM
ED3C: 60 1261 RTS
      1262
ED3D: 10 1263 CSST1 CLC
ED3E: AD 53 04 1264 LDA TEST53
ED41: 00 DE 1265 BNE CSST2
ED43: A9 68 1266 LDA #568
ED45: 85 3E 1267 STA A2L
ED47: AD 1F 1268 LDY #51F
ED49: AD 51 04 1269 LDA TEST51
ED4C: 00 02 1270 BNE #ETM1
ED4E: AD 35 1271 LDY #535
ED50: 60 1272 RETM1 RTS
      1273
ED51: A0 1A 1274 CSST2 LDY #51A
ED53: AD 51 04 1275 LDA TEST51
ED56: 00 02 1276 BNE CSST3
ED58: A0 33 1277 LDY #533
ED5A: A0 7D 04 1278 CSST3 LDA TEST7D
ED5D: 85 3C 1279 STA A1L
ED5F: 69 FF 1280 ADC #5FF
ED61: 85 3E 1281 STA A2L
ED63: AD 71 04 1282 LDA TEST71
ED66: 85 3D 1283 STA A1H
ED68: 69 0D 1284 ADC #50D
ED6A: 85 3F 1285 STA A2H
ED6C: CD 73 04 1286 CNP TEST73
ED6F: FD 18 1287 BEB CSST4
ED71: 9D 2B 1288 BCC #ETM2
ED73: A0 0F 1289 CSST4 LDY #50F
ED75: AD 51 04 1290 LDA TEST51
ED78: 00 02 1291 BNE CSST5
ED7A: A0 31 1292 LDY #531
ED7C: A0 73 04 1293 CSST5 LDA TEST73
ED7F: 85 3F 1294 STA A2H
ED81: A0 72 04 1295 LDA TEST72
ED84: 85 3E 1296 STA A2L
ED86: A9 01 1297 LDA #501
ED88: 8D 5D 04 1298 STA TEST5D

```

ED00:	A0	1299	RTS		
ED0C:	A0 10	1300	CSST6	LDY	#510
ED0E:	A0 51 04	1301		LDA	TEST51
ED11:	00 02	1302	RMC	CSST7	
ED13:	A0 32	1303		LDY	#532
ED15:	A0 72 04	1304	CSST7	LDA	TEST72
ED16:	C5 3E	1305		CMF	AZL
ED1A:	90 07	1306		BCC	CSST4
ED1C:	F0 05	1307		BEQ	CSST4
ED1E:	A0	1308	#ETK2	RTS	
		1309			
ED1F:	A0 03	1310	TFN	LDY	#503
ED41:	89 3C 00	1311	K0N	LDA:	A1L,Y
ED44:	99 70 04	1312		STA	TEST70,Y
ED47:	99 68 04	1313		STA	FOR1,Y
ED4A:	80	1314		DET	
ED4B:	10 F4	1315		BPL	K0N
ED4D:	60	1316		RTS	
		1317			
ED4E:	A0 95	1318	LOADT	LDY	#505
ED80:	89 60 04	1319	KOLMAN	LDA	RUF,Y
ED83:	99 74 04	1320		STA	TEST74,Y
ED86:	80	1321		DEY	
ED87:	10 F7	1322		BPL	KOLMAN
ED89:	C8	1323		IMY	
ED8A:	8C 50 04	1324		STY	TEST50
ED8D:	8C 53 04	1325		STY	TEST53
EDC0:	8C 51 04	1326		STY	TEST51
EDC3:	A2 FF	1327	INISUM	LDR	#0FF
EDC5:	86 2E	1328		STX	CHISUM
EDC7:	20 19 ED	1329		JSR	POP
EDCA:	A2 10	1330	PGW0N	LDR	#510
EDCC:	A0 25	1331	CPLTBIT	LDY	#525
EDCE:	20 F0 FC	1332		JSR	#0BIT
EDD1:	90 F7	1333		BCC	PGW0N
EDD3:	CA	1334		DEX	
EDD4:	D3 F6	1335		RMC	CPLTBIT
EDD4:	20 31 EE	1336		JSR	IMPW0N
EDD9:	20 FA FC	1337		JSR	#02BIT
EDDC:	A0 24	1338	LECL	LDY	#524
EDDE:	20 F0 FC	1339		JSR	#0BIT
EDD1:	80 F9	1340		BCL	LECL
EDD3:	20 F0 FC	1341		JSR	#0BIT
EDD6:	A0 3A	1342		LDY	#53A
EDD8:	20 2E	1343		JSR	LECHK
EDD8:	20 3D ED	1344		JSR	CSST1
EDDE:	20 2E	1345		JSR	LECHK
EDF1:	A0 53 04	1346		LDA	TEST53
EDF4:	F0 CD	1347		BEQ	INISUM
EDF6:	A0 3A	1348		LDY	#53A
EDF8:	20 EC FC	1349		JSR	#0BYTE
EDF8:	C5 2E	1350		CMF	CHISUM
EDFD:	F0 08	1351		BEQ	CHISUM
EDFF:	A9 A0	1352		LDA	#5A0
EDD1:	20 ED FD	1353		JSR	COUT
EDD4:	4C 20 FF	1354		JMP	PRERR
EDD7:	A5 3C	1355	OKCHSUM	LDA	A1L

;Inicializa CHISUM com 0FF
 ;Prepara para 1 byte
 ;Le meio ciclo
 ;E bit zero?
 ;Le 2 meios ciclos
 ;Le meio ciclo
 ;Le meio ciclo
 ;Le byte
 ;Le byte
 ;Le programa
 ;Imprime espaco
 ;Imprime erro

EE09: 00 70 04	1354	STA	TEST70	
EE0C: A5 30	1357	LDA	A18	
EE0E: 00 71 04	1358	STA	TEST71	
EE11: A0 50 04	1359	LDA	TEST50	
EE14: F0 A0	1360	BEQ	INTSUM	
EE14: 20 31	EE 1361	JSR	INPMAN	
EE19: A9 14	1362	LDA	#54	
EE1B: 05 24	1363	STA	CH	
EE1D: 4C 00 ED	1364	JMP	FDSAVE	
	344	PUT	PARTEN,02	
	31			
EE20: 20 EC FC	12	LEDM	JSR	BOBYTE ;Le programa e checa
EE23: 01 3C	13		STA	(A1L,X)
EE25: 45 2E	14		FOR	CHSUM
EE27: 05 2E	15		STA	CHSUM
EE29: 20 BA FC	16		JSR	NEXT#1
EE2C: A0 35	17		LDY	#535
EE2E: 90 F0	18		BCC	LEDM
EE30: A0	19		RTS	
	310			
EE31: A9 17	111	INPMAN	LDA	#517 ;Cursor na ultima linha
EE33: 05 25	112		STA	CV
EE35: A0 01	113		LDY	#501
EE37: 04 24	114		STY	CH
EE39: 00	115		DEC	
EE3A: 19 A0 04	116	CPLTHOME	LDA	BUF,Y ;Imprime o nome do programa
EE3D: 20 ED F0	117		JSR	COUT
EE40: C0	118		INY	
EE41: C0 06	119		CPY	#506
EE43: 00 F5	120		BNE	CPLTHOME
EE45: A9 A0	121	INPSP	LDA	#5A0 ;Imprime espaco
EE47: 20 ED F0	122		JSR	COUT
EE4A: 19 A0 04	123		LDA	BUF,Y ;Imprime comprimento do programa
EE4D: 20 0A F0	124		JSR	PRBYTE
EE50: C0	125		INY	
EE51: C0 08	126		CPY	#508
EE53: 00 F0	127		BNE	INPSP ;Imprime 2 bytes hex
EE55: A0 A7 04	128		LDA	BUF+7
EE5B: 00 20	129		BNE	RET#3
EE5A: A0 74 04	130		LDA	TEST74
EE5D: C9 A0	131		CMF	#5A0
EE5F: F0 0F	132		BEQ	INPMAN
EE61: A0 00	133		LDY	#100
EE63: 19 74 04	134	COMPNAME	LDA	TEST74,Y ;Compara nome do programa
EE66: 09 A0 04	135		CMF	BUF,Y
EE69: 00 1C	136		BNE	RET#3
EE6B: C0	137		INY	
EE6C: C0 06	138		CPY	#506
EE6E: 00 F3	139		BNE	COMPNAME
EE70: A9 01	140	INPMAN	LDA	#501
EE72: 00 53 04	141		STA	TEST53
EE75: A2 05	142		LDX	#505 ;Imprime " WAIT"
EE77: A0 03	143		LDY	#503
EE79: 20 F3 FA	144		JSR	MEMS
EE7C: A0 03	145		LDY	#503
EE7E: 19 A8 04	146	ATLCOP	LDA	PG1,Y
EE81: 99 70 04	147		STA	TEST70,Y

EEB4:	8B	148	DEY		
EEB5:	1D F7	149	BPL	ATLCON	;Atualiza comprimento do progr.
EEB7:	6D	150	RTS		
		151			
EEB8:	84 F8	152	TOSUB1	STY	COOL
EEBA:	64 34	153	LDY	YSAV	;Armazena pos.de CTRL em COOL
EEBC:	E6 34	154	IMC	YSAV	;Y-posicao do buffer DI
EEBE:	C9 06	155	CMP	MS66	;Verif. se e M (adv)
EEB9:	FD 0C	156	BEQ	ACMPAL	;Esta acompanhado de A ?
EEY2:	C9 E8	157	CMP	MS6B	;Verifica se e R (read)
EEY4:	FD 31	158	BEQ	ACMPA2	;Esta acompanhado de A ?
EEY6:	C9 FD	159	CMP	MSFD	;Verifica se e M (write)
EEY8:	FD 1F	160	BEQ	ACMPA3	;Esta acompanhado de A
EEYH:	C6 34	161	DEC	YSAV	;Volta posicao do buffer DI
EEYC:	00 4D	162	BNE	TSB	;TSB, se nao e posicao zero
EEYE:	89 00 02	163	ACMPAL	LDA	IN,Y
EEA1:	C9 C1	164	CMP	MSC1	;Verif.se caracter de IN, e A ?
EEA3:	00 06	165	BNE	VFFP	
EEA5:	2D 00 EC	166	JSR	MA	;Se e MA, seta prim.pag.de video
EEA8:	4C E7 EE	167	JMP	ZN001	
EEAB:	C9 00	168	CMP	MS00	;Compara com P
EEAD:	FD 04	169	BEQ	STMP	;Se NP, seta seg.pag.de video
EEAF:	C6 34	170	DEC	YSAV	;Volta posicao de IN
EEB1:	00 28	171	BNE	TSB	;Se nao posicao zero, TSB
EEB3:	2D F8 EB	172	STMP	JSR	NP
EEB6:	4C E7 EE	173	JMP	ZN001	;Beta segunda pag. de video
EEB9:	89 00 02	174	ACMPA3	LDA	IN,Y
EEBC:	C9 C1	175	CMP	MSC1	;Busca caracter de IN
EEBE:	FD 1E	176	BEQ	TSB	;Se for A,armazena no stack o endereço da rotina de WRITE
EEC0:	2D EF EE	177	JSR	KN000	;Senao armazena o end. de SAVEY
EEC3:	A2 01	178	LDX	MS01	coloca nome do progr.em BUF
EEC5:	00 0C	179	BNE	PHN000	
EEC7:	89 00 02	180	ACMPA2	LDA	IN,Y
EECA:	C9 C1	181	CMP	MSC1	;Se for A,armazena no stack o end. da rotina de READ
EECC:	FD 10	182	BEQ	TSB	
EECE:	2D EF EE	183	JSR	KN000	;Coloca nome do progr.em BUF
EED1:	A2 03	184	LDX	MS03	armazena o end. de L0A0T
EE03:	8D F8 FD	185	PHN000	LDA	SAVEEND,X
EE06:	48	186	PHA		
EE07:	CA	187	DEX		
EE0H:	8D F8 FD	188	LDA	SAVEEND,X	
EE0H:	48	189	PHA		
EE0C:	00 09	190	BNE	ZN001	
EE0E:	64 F8	191	TSB	LDY	COOL
EE0H:	A9 FE	192	TOSUB	LDA	MSFE
EE02:	48	193	PHA		
EE03:	89 E1 FF	194	LDA	SURTHL,Y	;End. baixo das rotinas
EE06:	48	195	PHA		
EE07:	A5 31	196	ZN001	LDA	MODE
EE09:	A0 00	197	ZN00	LDY	MS00
EE0H:	84 31	198	STY	MODE	
EE01:	6D	199	RTS		
		1100			
EE0E:	C8	1101	KN0001	INY	;Proximo caracter do buffer IN
EE0F:	89 00 02	1102	KN000	LDA	IN,Y
EEF2:	09 8D	1103	ORA	MS8D	;Transforma em NASCII
EEF4:	C9 AD	1104	CMP	MSAD	;Verifica se e espaco

EEF6:	FD F6	1105	BER	KWD01	;Sim, busca novo caracter	
EEF8:	C9 A2	1106	CMF	WSA2	;Verifica se e aspa	
EEFA:	00 01	1107	BNE	WASP		
EEFC:	C8	1108	INY		;Sim, incrementa buffer	
EEFD:	A2 00	1109	MOVSP	LOX	WS00	
EEFF:	ED 07	1110	KCHNAM	CPX	WS07	
EF01:	FD 19	1111	BER	ZBUFNAME	;E fim do BUF ?	
EF03:	89 00 02	1112	LDA	IN, Y		
EF06:	09 80	1113	ORA	WS80	;Transforma em NASCII	
EF08:	C9 80	1114	CMF	WS80	;Se RETURN zera BUF	
EF0A:	FD 10	1115	BER	ZBUFNAME		
EF0C:	C9 80	1116	CMF	WS80		
EF0E:	FD 0C	1117	BER	ZBUFNAME		
EF10:	C9 A2	1118	CMF	WSA2	;Se for aspa, completa com espaco	
EF12:	FD 07	1119	BER	CMPLSP		
EF14:	90 40 04	1120	STA	BUF, X	;Coloca caracter em BUF	
EF17:	E8	1121	INX			
EF18:	C8	1122	INY			
EF19:	00 E4	1123	BNE	KCHNAM		
EF1B:	C8	1124	CMPLSP	INY	;Proximo caracter de IN	
EF1C:	ED 08	1125	ZBUFNAME	CPX	WS08	
EF1E:	FD 08	1126	BER	FIMNAM	;Se fim do BUF	
EF20:	AF A0	1127	LDA	WSA0	salva posicao de IN e retorna	
EF22:	90 40 04	1128	STA	BUF, X	;Preenche BUF com espaco	
EF25:	E8	1129	INX			
EF26:	00 F4	1130	BNE	ZBUFNAME		
EF28:	04 34	1131	FIMNAM	STY	YSAV	
EF2A:	60	1132	RTS			
		1133				
EF2B:	20 7F FB	1134	BUSCON	JSR	P0CC	;Prepara indexacao para cor
EF2E:	AA	1135	TAX			
EF2F:	80 70 F4	1136	LDA	TARCSCR, X		;Busca byte de cor
EF32:	60	1137	RTS			
		1138				
EF33:	A0 59 C0	1139	EXTDIS	LDA	PRSTBN	;Printer stroke high
EF36:	A0 51 C0	1140		LDA	TOPER	;Preto e branco
EF39:	AF 20	1141	LDA	WS20		;Endereca primeira pag. de video
EF3B:	05 48	1142	STA	P0VA		
EF3D:	AF 05	1143	LDA	WS05		;Envia dado
EF3F:	80 08 C0	1144	STA	PRF2		
EF42:	A0 20	1145	L0T	WS20		;Gera um atrazo
EF44:	88	1146	LOOP20	DEC		
EF45:	00 FD	1147	BNE	LOOP20		
EF47:	A2 10	1148	LOX	WS10		;Loop de 500 vezes enquanto
EF49:	CA	1149	PRESF	DEX		;SC000 for diferente de 505
EF4A:	FD 08	1150	BER	FINESP		
EF4C:	A0 80 C0	1151	LDA	PRF1		
EF4F:	C9 05	1152	CMF	WS05		
EF51:	00 F6	1153	BNE	PRESF		
EF53:	AF 01	1154	LDA	WS01		
EF55:	00 02	1155	BNE	TENDANT		
EF57:	AF 80	1156	FINESP	LDA	WS80	
EF59:	05 47	1157	TENDANT	STA	P0VA	;Prepara P0VA com 0 ou 1
EF5B:	AF FF	1158	LDA	WSFF		
EF5D:	88 08 C0	1159	STA	PRF2		
EF60:	AF 80	1160	LDA	WS80		
EF62:	80 FD 9F	1161	STA	TSCANT		

EF65: A0 F0 9F	1162	LDA	TSCART	
EF66: F0 00	1163	BEQ	ZTSUTIL	
EF6A: A0 F1 9F	1164	LDA	INTCART	
EF6D: 85 06	1165	STA	LOCK	
EF6F: A0 F2 9F	1166	LDA	INTCART+1	
EF72: 85 07	1167	STA	LOCK+1	
EF74: A0 F0 9F	1168	LDA	TSCART	
EF77: 80 35 04	1169	ZTSUTIL	STA	ZTSUTIL
EF7A: 20 A8 C2	1170	JSR	TESTCAR	
EF7D: 60	1171	RTS		
	1172			
EF7E: 08	1173	CARRUF	PHP	;Salva status no stack
EF7F: 48	1174	FMA		;Salva acumulador no stack
EF80: C9 90	1175	CMF	#90	;Compara com CTRL-P
EF82: F0 12	1176	BEQ	ISCTP	;Se = CTRL-P
EF84: A5 47	1177	TEPISL	LDA	PEIL
EF86: 00 09	1178	BNE	NOVID	;Testa PEIL
EF88: 68	1179	PLA		;Recupera acumulador
EF89: 28	1180	PLP		;Recupera status
EF8A: 90 00 02	1181	STA	IN_X	;Armazena no buffer IN
EF8D: 08	1182	INX		;Proxima posicao do buffer
EF8E: 4C 54 F0	1183	JMP	BSROUT	
EF91: 68	1184	NOVID	PLA	;Recupera acumulador
EF92: 28	1185	PLP		;Recupera status
EF93: 4C 06 08	1186	JMP	PR56	
EF96: A0 41 04	1187	ISCTP	LDA	TEST41
EF99: C9 02	1188	CMF	#02	;Se \$44=0, -> TEPISL
EF9B: F0 07	1189	BEQ	TEPISL	
EF9D: 68	1190	PLA		;Recupera acumulador
EF9E: 28	1191	PLP		;Recupera status
EF9F: 4C 54 F0	1192	JMP	BSROUT	
EFA2: 08	1193	TEPISL	PHP	;Salva status no stack
EFA3: 48	1194	FMA		;Salva acumulador no stack
EFA4: A5 47	1195	LDA	PEIL	;Testa PEIL
EFA6: 00 05	1196	BNE	NOVID1	;Se diferente de 0
EFA8: 68	1197	PLA		;Recupera acumulador
EFA9: 28	1198	PLP		;Recupera status
EFAA: 4C 82 F0	1199	JMP	VFLIN	;Com status e acc intactos
EFA4: 68	1200	NOVID1	PLA	;Recupera acumulador
EFAE: 28	1201	PLP		;Recupera status
EFAF: 4C 0F 08	1202	JMP	PR5F	
EFB2: 08	1203	SWID	PHP	;Salva status no stack
EFB3: 48	1204	FMA		;Salva acumulador no stack
EFB4: A0 86 04	1205	LDA	TEST86	;Se \$48A=0 -> PR5CH
EFB7: C9 02	1206	CMF	#02	
EFB9: 00 05	1207	BNE	RECAP	;Se diferente de 0
EFBB: 68	1208	PLA		;Recupera acumulador
EFBC: 28	1209	PLP		;Recupera status
EFBD: 4C 0E F0	1210	JMP	PR5CH	
EFDD: 68	1211	RECAP	PLA	;Recupera acumulador
EFC1: 28	1212	PLP		;Recupera status
EFC2: 8E 80 04	1213	STX	TEST80	
EFC5: 4C 00 F0	1214	JMP	CHRTXT	
	1215			
EFCB: A5 26	1216	PLOT1	LDA	GBASL
EFCF: 85 26	1217	PLOT0	STA	GBASL
EFD0: 84 27	1218	STY	GBASH	;Le coordenada de linha
				;Armazena coordenada de linha
				;Armazena coordenada de coluna

EF0E: 0A	1219	TXA		;Salva X e Y no stack
EF0F: 4B	1220	PHA		
EF10: 9B	1221	TXA		
EF11: 4B	1222	PHA		
EF12: A5 30	1223	LDA COLOR		;Índice em X um valor de 0-7
EF14: 29 07	1224	AND #507		para buscar a cor dada por COLOR
EF16: AA	1225	TAX		
EF17: A5 27	1226	LDA BRASH		;Se a coluna a ser plot. é ímpar,
EF19: 4A	1227	LSR		;seta carry
EF1A: B0 AA FB	1228	LDA TARGON,X		;Carrega o byte a ser plotado
EF1D: 90 10	1229	BCC KIMWSEP		;Se a coluna for par, armazena
EF1F: 49 7F	1230	EOR #57F		o byte em RNDH,
EF21: 85 4F	1231	STA RNDH		senão inverte os bytes de imagem
EF23: 29 7F	1232	AND #57F		
EF25: C9 00	1233	CMP #500		;Se for cor branca ou preta,
EF27: F0 04	1234	BEQ DESINV		carrega em RNDH sem inverter
EF29: C9 7F	1235	CMP #57F		
EF2B: D0 04	1236	BNE LIMCORD		
EF2D: 49 7F	1237	DESINV		
EF2F: 85 4F	1238	EOR #57F		
EF31: A5 26	1239	KIMWSEP		
EF33: 20 03 FB	1240	LIMCORD		
EF35: 18	1241	LDA BRASH		;Coordenada da linha no acumul.
EF37: A5 0B	1242	CLC		;Prep.o end. base da microlinha 1
EF39: 45 27	1243	LDA #4L		
EF3B: 85 0B	1244	ADC BRASH		;Posiciona o end.da coluna em #4L
EF3D: A0 00	1245	STA #4L		
EF3F: A2 04	1246	LDY #500		;Zera indexador Y
F001: A5 4F	1247	LDX #504		;Prep.X para contar 4 loops
F003: 91 0B	1248	GRPLOT		;Coloca o dado na memória
F005: 1B	1249	LDA #4H		da página de vídeo
F007: A5 09	1250	CLC		
F009: 69 04	1251	LDA #4H		;Endereço a próxima microlinha
F00B: 85 09	1252	ADC #504		
F00D: CA	1253	STA #4H		
F00F: D0 F2	1254	DEX		;Repete 4 vezes
F011: 4B	1255	BNE GRPLOT		
F013: A8	1256	PLA		;Retorna Y e X do stack
F015: A8	1257	TAX		
F017: AA	1258	TAX		
F019: 60	1259	TAY		
	1260	BTS		
	1261	PUT PARTE21_02		
	1262			
F014: A9 80	12	IH3		
F016: 80 80 04	13	LDA #5B0		;Torna \$4B0 negativo
F019: 00 05	14	STA TESTB0		
F01B: A9 00	15	BNE SVYIS		;Desvia para SVYIS
F01D: 80 80 04	16	IN4		
F020: 9B	17	LDA #500		;Torna \$4B0 positivo
F021: 4B	18	STA TESTB0		
F022: A0 05	19	TXA		;Salva Y no stack
F024: A9 02	20	PHA		
F026: 20 0E 0A	21	PROCTOL		
F029: 20 43 F0	22	LDY #505		
F02C: EA 4E	23	ATBOUC		
F02E: 00 02	24	LDA #502		;Gera atraso p/ elim. bouncing
F030: E6 4F	25	JSR WAITB		
		JSR SCANI		
		INC RNDL		;Varre teclado
		BNE RNDH		;Incrementa o par (RND)
		INC RNDH		

F032:	2C 80 04	116	NRINC	RIT	TESTB0	;Testa S400
F035:	30 04	117		RRI	TESTCL	;Se entrou por J03,TESTCL
F037:	80 E9	118		RCS	PROCTOL	;Se tem tecla apertada, PROCTOL
F039:	90 02	119		BCC	NOY	;Se nao tem tecla apertada,NOY
F03B:	90 E5	120	TESTCL	BCC	PROCTOL	;Se nao tem tecla apertada
F03C:	88	121	NOY	OEY		;Decrementa ate a quinta vez
F03E:	00 E4	122		RNE	ATEUAC	;Volta para testar teclado
F040:	68	123		PLA		;Salva Y no stack
F041:	68	124		TAY		
F042:	60	125		RTS		
		126				
F043:	80 SE C0	127	SCANS	STA	BTCHTD	;Reseta bit CONTROL
F046:	0A	128		TXA		;Salva X e Y no stack
F047:	48	129		PHA		
F048:	98	130		TXA		
F049:	48	131		PHA		
F04A:	AF 00	132		LDA	MS00	;Zera GRASH
F04C:	85 27	133		STA	GRASH	;Como contador de caracteres
F04E:	AF 01	134		LDA	MS01	;Habilita linha I da matriz
F050:	85 26	135		STA	GRASL	
F052:	A2 08	136		LDX	MS08	;Prepara X para contar 8 linhas
F054:	A5 26	137	HAMLIN	LDA	GRASL	
F056:	80 00 C0	138		STA	XROUT	;Envia dado para o teclado
F059:	A0 10 C0	139		LDA	INTFIN	;Le dado do teclado (resposta)
F05C:	A0 06	140		LDR	MS06	;Prepara Y para contar 6 colunas
F05E:	4A	141	NOCOL	ROR		;Testa bit atraves do carry
F05F:	90 03	142		BCC	NOKEY	;Nao e a tec.apert.,novo teste
F061:	20 81 F0	143		JSR	IDFSCTL	;Se encontramos tecla apertada
F064:	E6 27	144	NOKEY	INC	GRASH	;Decrem. contador do caracter
F066:	88	145		DEY		;Decrementa coluna
F067:	00 F5	146		RNE	NOCOL	;Se nao foi a ultima coluna
F069:	18	147		CLC		;Limpa carry
F06A:	26 26	148		ROL	GRASL	;Habilita proxima linha
F06C:	CA	149		DEX		;Decrementa contador de linha
F06D:	00 E5	150		RNE	HAMLIN	;Nao foi a ultima linha,repita
F06F:	18	151		CLC		;Acabou teste, limpa carry
F070:	68	152	SHVYX	PLA		;Recupera Y e X do stack
F071:	68	153		TAY		
F072:	68	154		PLA		
F073:	6A	155		TAX		
F074:	AF 00	156	CLXKEY	LDA	MS00	;limpa registrador XROUT
F076:	80 00 C0	157		STA	XROUT	
F079:	90 03	158		BCC	BT1	
F07B:	20 85 F0	159		JSR	SEPTAB	
F07E:	40	160	BT1	RTS		
F07F:	68	161	BT2	PLA		;Recupera a resposta do teclado
F080:	60	162		RTS		
		163				
F081:	48	164	IDFSCTL	PHA		;Salva resp.do teclado no stack
F082:	A5 27	165		LDA	GRASH	;Se e tecla SHIFT
F084:	F0 F9	166		BEQ	BT2	;Retorna e continua teste
F086:	68	167		PLA		;Retira a resposta do stack
F087:	A9 00	168		LDA	MS00	;limpa XROUT
F089:	80 00 C0	169		STA	XROUT	
F08C:	A9 01	170		LDA	MS01	;Habilita a linha I
F08E:	80 00 C0	171		STA	XROUT	
F091:	A0 10 C0	172		LDA	INTFIN	

F094: 6A	173	ROR	:Testa col. 1, linha 1 (SHIFT)
F095: 90 06	174	BCC SETCTL	:Se nao e SHIFT
F097: A5 27	175	LDA GRASH	:Seta bit 7 do cont. de carac.
F099: 0F 80	176	ORA #580	
F09B: 05 27	177	STA GRASH	
F09D: A9 00	178	SETCTL LDA #500	:Limpa KBOUT
F09F: 80 00 C0 179		STA KBOUT	
F0A2: 80 5F C0 180		STA BTOUT1	:Seta CONTROL
F0A5: A0 10 C0 181		LDA KSTFIN	:Le KBIN
F0A8: 6A	182	ROR	:Testa tecla control
F0A9: 90 06	183	BCC SCWEND	:Mao e CONTROL, va para SCWEND
F0AB: A5 27	184	LDA GRASH	:seta bit 6
F0AD: 09 40	185	ORA #540	:do contador de caracteres
F0AF: 85 27	186	STA GRASH	
F0B1: 68	187	SCWEND PLA	
F0B2: 68	188	PLA	
F0B3: 38	189	SEC	:Seta carry
F0B4: 80 BA	190	BCC SAVE1	
F0B6: B4 24	191	PROCV0 STY CH	:Posiciona cursor horizontal
F0B8: 48	192	PRCV01 PHA	:Salva acumulador no stack
F0B9: C9 80	193	CMF #580	:E RETURN
F0BB: FD 40	194	BEQ FIM1	:Sim,retorna a rotina principal
F0BD: 4C B2 CF 195		JMP SVID	:Mao,rotina de saida de dados
F0BD: A2 F2	196	CHMTXT LOX #160CHN	:Endereco alt.da tab.de geracao
F0C2: 86 07	197	STX R3H	de caracteres de texto em R3H
F0C4: C9 80	198	CMF #580	:Se for ASCII -> TEXTO
F0C4: 80 06	199	BCC TEXTO	
F0C6: E9 00	100	SBC #500	
F0CA: A2 F4	101	LOX #160GRF	:Endereco alt.da tab.de geracao
F0CC: 86 07	102	STX R3H	de caracteres graficos em R3H
F0CD: AE 80 DA	103	TEXT0 LOX TEST80	
F0D1: 29 3F	104	AND #53F	:Seta valor entre 0 e 63
F0D3: DA	105	ASL	:Multiplica por 8
F0D4: DA	106	ASL	
F0D5: DA	107	ASL	
F0D6: 85 DA	108	STA R3L	
F0D8: A5 07	109	LDA R3H	
F0DA: 69 00	110	AOC #500	
F0DC: 85 07	111	STA R3H	
F0DE: 20 F3 F8	112	POSCN JSR CVID	:Mult .CV por 8 ,resul .em DELN
F0E1: BA	113	TXA	:Salva X e Y no stack
F0E2: 48	114	PHA	
F0E3: 98	115	TYA	
F0E4: 48	116	PHA	
F0E4: A2 08	117	LOX #500	:X como contador
F0E7: A0 00	118	LDY #500	de 8 microlinhas de video
F0E9: 20 F3 FC	119	IMPCH JSR EPVID	:prepara R4 como endereco base
F0EC: 18	120	CLC	: da pagina de video
F0ED: A5 08	121	LDA R4L	:Posiciona endereco de video
F0EF: A5 24	122	AOC CH	para coluna desejada
F0F1: 85 08	123	STA R4L	
F0F3: A5 09	124	LDA R4H	
F0F5: 69 00	125	AOC #500	
F0F7: 85 09	126	STA R4H	
F0F9: B1 06	127	LDA (R3L),Y	:Pega RYTE na tab.de enderecos
F0FB: A5 32	128	EDM INVFLS	:Modo normal ou inverso?
F0FD: 91 08	129	STA (R4L),Y	:Seta pagina de video

F0FF:	E6 E3	1130	INC	DELH	;Proxima microlinha
F101:	E6 06	1131	INC	R3L	;Proxima byte da tabela
F103:	CA	1132	DEX		;Microlinha 0 ?
F104:	00 E3	1133	RMC	EMPOH	;Mas, imprime proxima microlinha
F106:	68	1134	PLA		;Recup.T e X do stack e retorna
F107:	68	1135	TAY		
F108:	68	1136	PLA		
F109:	AA	1137	TAX		
F10A:	68	1138	FIN1	PLA	
F10B:	60	1139	RTS		
		1140			
F10C:	90 00 04	1141	ZPS4	STA	P54,X
F10F:	EB	1142	INC		
F110:	00 FA	1143	RMC	ZPS4	
F112:	A9 20	1144	LDA	W520	;P54H=420, seta pag 1 de video
F114:	85 48	1145	STA	P54H	;Seta pag. de video
F116:	60	1146	RTS		
		1147			
F117:	40 82 04	1148	COOCC	LDA	FINL18
F11A:	00 0E	1149	RMC	TSA44	;se FINL18 =10 va para TSA44
F11C:	40 41 04	1150	LDA	TEST41	
F11F:	F0 06	1151	BEQ	JNCR	;se \$441=0 fazer na CR
F121:	68	1152	TAY		;Y=8
F122:	A6 25	1153	LDR	CV	;Z=CV
F124:	20 00 C2	1154	JSR	EMPR	
F127:	4C 62 FC	1155	JNCR	JMP	CR
F12A:	25 46	1156	TSA44	LDA	A46
F12C:	C9 03	1157	CMR	W503	;Se A46=0 JMP W7006
F12E:	F0 03	1158	BEQ	JCAR16	;Se A46<13 JMP \$815
F130:	4C 15 08	1159	JMP	PRO15	
F133:	4C 06 90	1160	JMP	CAR16	
		346	PUT	PART2J,02	
F136:	A5 24	11	SCROLL	LDA	CH
F138:	48	12	PHA		;Salva posicao de CH no stack
F139:	A5 22	13	LDA	MMOTOP	
F13B:	85 06	14	STA	R3L	;Coordenada do topo da tela em R3L
F13D:	E6 06	15	INC	R3L	
F13F:	20 F5 F8	16	JSR	A38	;Prepara DELH
F142:	20 73 FC	17	JSR	EPOH10	;Calcula o endereco base da
F145:	A4 08	18	LDR	R4L	pag de video em R4, a partir
F147:	A5 09	19	LDA	R4H	de DELH
F149:	18	110	CLC		
F14A:	69 20	111	ADC	W520	;Prepara BASL e BASH com o end.
F14C:	AA	112	TAX		de mem.correspo.a microlinha 1
F14D:	38	113	SCRL1	SEC	do topo da tela
F14E:	84 28	114	STY	BASL	;Calcula o endereco fisico de
F150:	0A	115	TXA		memoria correspondente a
F151:	E9 20	116	SBC	W520	microlinha 1 da proxima
F153:	85 29	117	STA	BASH	linha de texto
F155:	98	118	TYA		
F156:	69 7F	119	ADC	W57F	
F158:	A8	120	TAY		
F159:	8A	121	TXA		
F15A:	E9 1F	122	SBC	W51F	
F15C:	AA	123	TAX		
F15D:	29 04	124	AND	W504	
F15F:	F0 0A	125	BEQ	SCRL2	

F161: 90	126	TYA	
F162: C9 50	127	CHP #550	
F164: F0 32	128	BEQ CLRVL	
F166: 49 28	129	ADC #525	
F168: A0	130	TAY	
F169: A6 40	131	_LDX P0VH	
F16B: 06 28	132	STX BAS2H	;Armazena o endereço em
F16D: 04 28	133	STY BAS2L	;BAS2L e BAS2H
F16F: A2 00	134	LOX #500	;Prep. contador de 8 microlinhas
F171: A4 20	135	LDY WMDLFT	;Mo início da linha horizontal
F173: B1 28	136	LDX (BAS2L),Y	;transf.a microlinha de vídeo
F175: 91 28	137	STA (BASL),Y	de (BAS2L) para (BASL)
F177: C8	138	DNY	
F179: C4 21	139	CPY WMDMTH	;Fim da linha horizontal ?
F17A: 00 F7	140	BNE SCRL4	;Continua transferindo a linha
F17C: 18	141	CLC	;transfere a proxima microlinha
F17D: A9 04	142	LDA #504	ate a microlinha 8
F17F: 45 28	143	ADC BAS2H	
F181: 05 28	144	STA BAS2H	
F183: A9 04	145	LDA #504	
F185: 45 29	146	ADC BASH	
F187: 05 29	147	STA BASH	
F189: CA	148	DEX	
F18A: 00 E5	149	BNE SCRL3	;Se nao e a microlinha 8
F18C: A4 2A	150	LDY BAS2L	
F18E: A6 28	151	LDX BAS2H	
F190: E6 06	152	INC #3L	;Desce uma linha de texto
F192: A5 06	153	LDA #3L	
F194: C5 23	154	CHP WMDMTH	;E ultima linha de texto?
F196: 00 85	155	BNE SCRL1	;Nao, proxima linha de texto
F198: A0 00	156	LDY #500	;limpa a ultima linha de texto
F19A: 20 96 FC	157	JSR CLEOL2	
F19D: A6 23	158	LOX WMDMTH	
F19F: CA	159	DEX	
F1A0: 06 25	160	STX CV	;Posiciona o CV na ultima linha
F1A2: 68	161	PLA	;Retorna a posicao do CH
F1A3: 85 24	162	STA CH	
F1A5: 60	163	RTS	
	164		
F1A6: 4C 0A EC	165	CONT2 JMP CONT3	;Saida de dados pelo video
F1A9: 04 35	166	STY TSWVL	;Salva Y
F1AB: 06 4E	167	STX RMDL	;Salva X
F1AD: 49	168	PNA	;Salva acumulador na stack
F1AE: 23 7D F8	169	JSR VIDWMT	;primeiro caracter
F1B1: 68	170	PLA	;Recupera acumulador do stack
F1B2: C9 80	171	CHP #500	;Verifica se e um RETURN
F1B4: F0 11	172	BEQ SETCR	;Se for RETURN -> SETCR
F1B6: C9 90	173	CHP #590	;E CONTROL-P ?
F1B8: 00 00	174	BNE NTCTP	;Nao, -> NTCTP
F1BA: 20 82 F8	175	JSR POSCUR	
F1BD: A9 00	176	LDA #500	
F1BF: 4C E3 F1	177	JMP ESPERA	
F1C2: A6 4E	178	NTCTP LOX RMDL	
F1C4: A4 35	179	LDY TSWVL	
F1CA: A0	180	RTS	
	181		
F1C7: 4C 0A EC	182	SETCR JMP TESFL6	

F1C4:	08	>83	CONTOR	PHP		;Salva status no stack
F1C8:	A5 46	184		LDA	A46	;Se A46=3 -> SF1F5
F1C0:	C9 00	185		CMP	#503	
F1C7:	F0 24	186		0ER	JCART9	
F101:	C9 04	187		CMP	#504	;Se A46=4 -> SF1F9
F103:	F0 24	188		0ER	JPRG18	
F105:	20	189		PLP		;Recupera status do stack
F104:	A6 25	190		LDX	CV	;Carrega X com o conteúdo de CV
F108:	AC 42 04	191		L0Y	TEST42	
F109:	98	192		TYA		
F10C:	40	193		PHA		
F1D0:	CA	194	DECKY	DEX		
F10E:	88	195		DEY		
F10F:	00 FC	196		0NE	DECKY	
F1E1:	60	197		PLA		;Retorna Y do stack
F1E2:	A8	198		TAY		
F1E3:	48	199	ESPERA	PHA		
F1E4:	AF 7F	1100		LDA	#57F	;Berra um atrazo
F1E6:	20 A8 FC	1101		JSR	WAIT	
F1E9:	68	1102		PLA		
F1EA:	20 00 C2	1103		JSR	IMPR	
F1ED:	A0 01	1104		L0Y	#501	;Carrega \$442 com #01
F1EF:	8C 42 04	1105		STY	TEST42	
F1F2:	4C C2 F1	1106		JMP	HTCTP	;Recupera X e Y e retorna
F1F5:	20	1107	JCART9	PLP		
F1F6:	4C 0F 9D	1108		JMP	CART9	
F1F9:	28	1109	JPRG18	PLP		
F1FA:	4C 18 08	1110		JMP	PRG18	
F1FD:	FF FF FF	1111		HEX	FFFFFF	

*Tabela de geracao de caracteres de texto

F200:	00 1C 22	1115	GETCHR	HEX	001C2220A1A023C	; @
F203:	2A 3A 1A 02 3C			HEX	000B142223E2322	; A
F208:	00 08 14	1116		HEX	001E22221E22221E	; B
F210:	00 1E 22	1117		HEX	001C222020202221C	; C
F213:	22 1E 22 22 1E			HEX	001E22222222221E	; D
F218:	00 1C 22	1118		HEX	003E2021E02023E	; E
F21B:	02 02 02 22 1C			HEX	003E2021E02023E	; F
F220:	00 1E 22	1119		HEX	00302020232223C	; G
F223:	22 22 22 22 1E			HEX	00222223E222222	; H
F228:	00 3E 02	1120		HEX	001C0B0808081C	; I
F22B:	02 1E 02 02 3C			HEX	0022120A0A0A122	; K
F230:	00 3E 02	1121		HEX	000202020202023E	; L
F233:	02 1E 02 02 02					
F236:	00 3C 02	1122				
F239:	02 02 3C 22 3C					
F240:	00 22 22	1123				
F243:	22 3C 22 22 22					
F246:	00 1C 00	1124				
F249:	08 08 08 08 1C					
F250:	00 20 20	1125				
F253:	20 20 20 22 1C					
F256:	00 22 12	1126				
F259:	0A 0A 0A 12 22					
F260:	00 02 02	1127				

F263:	02 02 02 02 3E	
F268:	00 22 36 1129	HEX 0022362A2A222222 ; H
F269:	2A 2A 22 22 22	
F270:	00 22 22 1129	HEX 0022222A2A222222 ; H
F273:	2A 2A 22 22 22	
F278:	00 1C 22 1130	HEX 001C22222222221C ; 0
F279:	22 22 22 22 1C	
F280:	00 1E 22 1131	HEX 001E22221E020202 ; F
F283:	22 1E 02 02 02	
F288:	00 1C 22 1132	HEX 001C2222222A122C ; 6
F289:	22 22 2A 12 2C	
F290:	00 1E 22 1133	HEX 001E22221E0A1222 ; 8
F293:	22 1E 0A 12 22	
F298:	00 1C 22 1134	HEX 001C22021C20221C ; 5
F299:	02 1C 20 22 1C	
F2A0:	00 3E 08 1135	HEX 003E080808080808 ; T
F2A3:	08 08 08 08 08	
F2A8:	00 22 22 1136	HEX 002222222222221C ; U
F2AB:	22 22 22 22 1C	
F2B0:	00 22 22 1137	HEX 0022222222221408 ; V
F2B3:	22 22 22 14 08	
F2B8:	00 22 22 1138	HEX 002222222A2A3622 ; W
F2BB:	22 2A 2A 36 22	
F2C0:	00 22 22 1139	HEX 0022221408142222 ; X
F2C3:	14 08 14 22 22	
F2C8:	00 22 22 1140	HEX 0022221408080808 ; Y
F2CB:	14 08 08 08 08	
F2D0:	00 3E 20 1141	HEX 003E20100804023E ; Z
F2D3:	10 08 04 02 3E	
F2D8:	00 1F 06 1142	HEX 001F06060A06061F ;
F2DB:	06 06 06 06 1F	
F2E0:	00 00 02 1143	HEX 0000020408102000 ; /
F2E3:	04 08 10 20 00	
F2E8:	00 7C 30 1144	HEX 007C30303030307C ;
F2EB:	30 30 30 30 7C	
F2F0:	00 00 00 1145	HEX 0000000814200000 ; *
F2F3:	08 14 22 00 00	
F2F8:	00 00 00 1146	HEX 000000000000003E ;
F2FB:	00 00 00 00 3E	
F300:	00 00 00 1147	HEX 0000000000000000 ; Espaco
F303:	00 00 00 00 00	
F308:	00 08 08 1148	HEX 0008000800000008 ; !
F30B:	08 08 08 00 08	
F310:	00 14 14 1149	HEX 0014141400000000 ; "
F313:	14 00 00 00 00	
F318:	00 14 14 1150	HEX 0014143E143E1414 ; #
F31B:	3E 14 3E 14 14	
F320:	00 18 78 1151	HEX 0018781438383C10 ; \$
F323:	14 38 38 3C 10	
F328:	00 0A 26 1152	HEX 000A261008043220 ; %
F32B:	10 00 0A 32 30	
F330:	00 08 14 1153	HEX 0008141408542458 ; &
F333:	14 00 54 24 58	
F338:	00 10 10 1154	HEX 0010101000000000 ; '
F33B:	10 00 00 00 00	
F340:	00 08 04 1155	HEX 0008040202020408 ; (
F343:	02 02 02 04 08	

```

F340: 00 00 10 1154      HEX 0000102020201000 ; 1
F340: 20 20 20 10 00
F350: 00 00 2A 1157      HEX 00002A1C001C2A00 ; 2
F353: 1C 00 1C 2A 00
F350: 00 00 00 1150      HEX 0000000030000000 ; 3
F350: 00 3E 00 00 00
F360: 00 00 00 1159      HEX 0000000000000004 ; 4
F363: 00 00 00 00 04
F360: 00 00 00 1160      HEX 0000000030000000 ; 5
F360: 00 3E 00 00 00
F370: 00 00 00 1161      HEX 0000000000000000 ; 6
F373: 00 00 00 00 00
F370: 00 00 20 1162      HEX 0000201000040200 ; 7
F370: 10 00 04 02 00
347
F380: 00 1C 22 11      PUT PARTE2L_02
F383: 32 2A 2A 22 1C      HEX 001C22322A2A221C ; 0
F380: 00 00 0C 12      HEX 00000C000000001C ; 1
F380: 00 00 00 00 1C
F390: 00 1C 22 13      HEX 001C22201004020E ; 2
F393: 20 10 04 02 3E
F390: 00 3E 20 14      HEX 003E20101020221C ; 3
F390: 10 10 20 22 1C
F3A0: 00 10 10 15      HEX 00101014120E1010 ; 4
F3A3: 14 12 3E 10 10
F3A0: 00 3E 02 16      HEX 003E021E2020221C ; 5
F3A0: 1E 20 20 22 1C
F3B0: 00 30 04 17      HEX 003004021E22221C ; 6
F3B3: 02 1E 22 22 1C
F3B0: 00 3E 20 10      HEX 003E201000040404 ; 7
F3B0: 10 00 04 04 04
F3C3: 00 1C 22 19      HEX 001C22221C22221C ; 8
F3C3: 22 1C 22 22 1C
F3C0: 00 1C 22 10      HEX 001C22222020100E ; 9
F3C0: 22 3E 20 10 0E
F3D0: 00 00 00 111      HEX 0000000000000000 ; 10
F3D3: 00 00 00 00 00
F3D0: 00 00 00 112      HEX 0000000000000004 ; 11
F3D0: 00 00 00 00 04
F3E3: 00 10 00 113      HEX 0010000402040010 ; 12
F3E3: 04 02 04 00 10
F3E0: 00 00 00 114      HEX 0000000000000000 ; 13
F3E0: 3E 00 3E 00 00
F3F3: 00 04 00 115      HEX 0004001020100004 ; 14
F3F3: 10 20 10 00 04
F3F0: 00 1C 22 116      HEX 001C221000000000 ; 15
F3F0: 10 00 00 00 00

```

117

118

119

*Tabela de geracao de caracteres graficos

```

F400: 00 00 00 120      GERBRF HEX 000000000000007F
F403: 00 00 00 00 7F
F400: 00 00 00 121      HEX 0000000000007F7F
F400: 00 00 00 7F 7F
F410: 00 00 00 122      HEX 000000007F7F7F7F
F413: 00 00 7F 7F 7F
F410: 00 00 00 123      HEX 000000007F7F7F7F

```

F418:	00 7F 7F 7F	
F420:	00 00 00 124	HEX 0000007F7F7F7F
F421:	7F 7F 7F 7F	
F426:	00 00 7F 125	HEX 00007F7F7F7F7F
F428:	7F 7F 7F 7F	
F430:	00 7F 7F 126	HEX 007F7F7F7F7F7F
F431:	7F 7F 7F 7F	
F438:	01 01 01 127	HEX 01010101010101
F439:	01 01 01 01	
F440:	03 03 03 128	HEX 03030303030303
F441:	03 03 03 03	
F448:	07 07 07 129	HEX 07070707070707
F449:	07 07 07 07	
F450:	0F 0F 0F 130	HEX 0F0F0F0F0F0F0F
F451:	0F 0F 0F 0F	
F458:	1F 1F 1F 131	HEX 1F1F1F1F1F1F1F
F459:	1F 1F 1F 1F	
F460:	3F 3F 3F 132	HEX 3F3F3F3F3F3F3F
F461:	3F 3F 3F 3F	
F468:	7F 7F 7F 133	HEX 7F7F7F7F7F7F7F
F469:	7F 7F 7F 7F	
	134	
F470:	00 55 2A 135	TAMCSCR HEX 00552A7FB005A9FF ;Tabela usada por SCW
F471:	7F 80 05 AA FF	
	136	
	137	*Continuacao da tabela de geracao de caracteres graficos
	138	
F478:	00 00 7F 139	HEX 00007F00007F0000
F479:	00 00 7F 00 00	
F480:	08 08 79 140	HEX 0808780808780808
F481:	08 08 78 08 00	
F488:	08 08 7F 141	HEX 08087F08087F0808
F489:	08 08 7F 08 08	
F490:	08 08 0F 142	HEX 08080F08080F0808
F491:	08 08 0F 08 08	
F498:	00 40 60 143	HEX 0040600040607C7E7F
F499:	70 78 7C 7E 7F	
F4A0:	00 01 03 144	HEX 000103070F1F3F7F
F4A1:	07 0F 1F 3F 7F	
F4A8:	00 7F 7E 145	HEX 007F7E7C78706A040
F4A9:	7C 78 70 60 40	
F4B0:	00 7F 3F 146	HEX 007F3F1F0F070301
F4B1:	1F 0F 07 03 01	
F4B8:	7F 00 00 147	HEX 7F00000000000000
F4B9:	00 00 00 00 00	
F4C0:	00 00 00 148	HEX 000000007F000000
F4C1:	00 7F 00 00 00	
F4C8:	08 08 08 149	HEX 0808080808080808
F4C9:	08 08 08 08 08	
F4D0:	10 10 10 150	HEX 1010101010101010
F4D1:	10 10 10 10 10	
F4D8:	00 40 20 151	HEX 0040201000040201
F4D9:	10 08 04 02 01	
F4E0:	00 01 02 152	HEX 0001020408102040
F4E1:	04 08 10 20 40	
F4E8:	00 1C 3E 153	HEX 001C3E3E3E3E1C00
F4E9:	3E 3E 3E 1C 00	

F4F0:	00 1C 22 154	HEX	001C22222221C00	
F4F3:	22 22 22 1C 00			
F4F8:	08 08 08 155	HEX	080808087000000	
F4FB:	08 7F 00 00 00			
F500:	00 00 00 156	HEX	000000007F00000	
F503:	00 7F 08 08 08			
F506:	08 08 08 157	HEX	080808080F00000	
F509:	08 0F 08 08 08			
F510:	08 08 08 158	HEX	080808087000000	
F513:	08 78 08 08 08			
F518:	00 08 1C 159	HEX	00081C3E7F7F1C3E	
F51B:	3E 7F 7F 1C 3E			
F520:	00 36 7F 160	HEX	00367F7F7F3E1C08	
F523:	7F 7F 3E 1C 08			
F526:	00 68 1C 161	HEX	00081C3E7F3E1C08	
F529:	3E 7F 3E 1C 08			
F530:	00 1C 1C 162	HEX	001C1C7F7F68083E	
F533:	7F 7F 68 08 3E			
F538:	00 00 00 163	HEX	000000007800000	
F53B:	00 78 08 08 08			
F540:	00 00 00 164	HEX	000000000F00000	
F543:	00 0F 08 08 08			
F546:	08 08 08 165	HEX	0808087800000000	
F54B:	78 00 08 00 00			
F550:	08 08 08 166	HEX	0808080F00000000	
F553:	0F 00 00 00 00			
F558:	00 00 00 167	HEX	00000000A0100008	
F55B:	00 60 10 08 08			
F560:	00 00 00 168	HEX	00000000000040808	
F563:	00 03 04 08 08			
F568:	08 08 10 169	HEX	0808106000000000	
F56B:	60 00 00 00 00			
F570:	08 08 04 170	HEX	0808040000000000	
F573:	03 00 00 00 00			
F578:	41 22 14 171	HEX	4122140808142241	
F57B:	08 08 14 22 41			
F580:	00 08 08 172	HEX	000808007F000808	
F583:	00 7F 00 08 08			
F588:	08 08 08 173	HEX	080808087F080808	
F58B:	08 7F 08 08 08			
	174			
F590:	20 FF 00 175	DISK10	HEX	20FF00FF03FF3C
F593:	FF 03 FF 3C			
	176			
F597:	80 80 80 177	TITULO	ASC	"0002KT"
F59A:	82 C8 84			
	178			
	179	*Tabela de codigos do teclado		
	180			
F59D:	FF 181	TABASC	HEX	FF
F59E:	FF 182		HEX	FF
F59F:	FF 183		HEX	FF
F5A0:	FF 184		HEX	FF
F5A1:	C2 185		HEX	C2
F5A2:	82 186		HEX	82
F5A3:	22 187		HEX	22
F5A4:	12 188		HEX	12

; Espaço reservado sem tecla
 ; 0
 ; CTRL-0
 ; SHIFT-0
 ; SHF/CTR-0

FSAS: 06	189	HEX 06	: V
FSAG: 96	190	HEX 96	: CTRL-V
FSAN: 2B	191	HEX 2B	: SHFT-V
FSAB: 18	192	HEX 18	: SHF/CTR-V
FSAP: C3	193	HEX C3	: C
FSAA: 83	194	HEX 83	: CTRL-C
FSAB: 2A	195	HEX 2A	: SHFT-C
FSAC: 1A	196	HEX 1A	: SHF/CTR-C
FSAD: 08	197	HEX 08	: X
FSAE: 98	198	HEX 98	: CTRL-X
FSAF: 2F	199	HEX 2F	: SHFT-X
FSB0: 17	1100	HEX 17	: SHF/CTR-X
FSB1: 0A	1101	HEX 0A	: Z
FSB2: 9A	1102	HEX 9A	: CTRL-Z
FSB3: 2E	1103	HEX 2E	: SHFT-Z
FSB4: 16	1104	HEX 16	: SHF/CTR-Z
FSB5: FF	1105	HEX FF	: Espaço reservado sem tecla
FSB6: FF	1106	HEX FF	
FSB7: FF	1107	HEX FF	
FSB8: FF	1108	HEX FF	
FSB9: C7	1109	HEX C7	: S
FSBA: 87	1110	HEX 87	: CTRL-S
FSBB: 20	1111	HEX 20	: SHFT-S
FSBC: 10	1112	HEX 10	: SHF/CTR-S
FSBD: C6	1113	HEX C6	: F
FSBE: 86	1114	HEX 86	: CTRL-F
FSBF: 29	1115	HEX 29	: SHFT-F
FSB0: 19	1116	HEX 19	: SHF/CTR-F
FSB1: C4	1117	HEX C4	: D
FSB2: 84	1118	HEX 84	: CTRL-D
FSB3: 28	1119	HEX 28	: SHFT-D
FSB4: 18	1120	HEX 18	: SHF/CTR-D
FSB5: 03	1121	HEX 03	: S
FSB6: 93	1122	HEX 93	: CTRL-S
FSB7: 20	1123	HEX 20	: SHFT-S
FSB8: 15	1124	HEX 15	: SHF/CTR-S
FSB9: C1	1125	HEX C1	: A
FSBA: 81	1126	HEX 81	: CTRL-A
FSBB: 2C	1127	HEX 2C	: SHFT-A
FSBC: 14	1128	HEX 14	: SHF/CTR-A
FSBD: A0	1129	HEX A0	: "Espaço"
FSBE: A0	1130	HEX A0	: CTRL-Espaço
FSBF: A0	1131	HEX A0	: SHFT-Espaço
FSB0: 34	1132	HEX 34	: SHF/CTR-Espaço
FSB1: 04	1133	HEX 04	: T
FSB2: 94	1134	HEX 94	: CTRL-T
FSB3: 1E	1135	HEX 1E	: SHFT-T
FSB4: 0C	1136	HEX 0C	: SHF/CTR-T
FSB5: 02	1137	HEX 02	: R
FSB6: 92	1138	HEX 92	: CTRL-R
FSB7: 27	1139	HEX 27	: SHFT-R
FSB8: 08	1140	HEX 08	: SHF/CTR-R
FSB9: C5	1141	HEX C5	: E
FSBA: 85	1142	HEX 85	: CTRL-E
FSBB: 26	1143	HEX 26	: SHFT-E
FSBC: 0A	1144	HEX 0A	: SHF/CTR-E
FSBD: 07	1145	HEX 07	: M

F50E:	97	1146	HEX 97	:	CTRL-M
F50F:	25	1147	HEX 25	:	SHIFT-M
F510:	09	1148	HEX 09	:	SHF/CTR-M
F511:	01	1149	HEX 01	:	4
F512:	91	1150	HEX 91	:	CTRL-N
F513:	24	1151	HEX 24	:	SHIFT-N
F514:	08	1152	HEX 08	:	SHF/CTR-N
F515:	88	1153	HEX 88	:	(-)
F516:	88	1154	HEX 88	:	CTRL (-)
F517:	88	1155	HEX 88	:	SHIFT (-)
F518:	3C	1156	HEX 3C	:	SHF/CTR (-)
F519:	85	1157	HEX 85	:	5
F51A:	85	1158	HEX 85	:	CTRL-5
F51B:	45	1159	HEX 45	:	SHIFT-5
F51C:	05	1160	HEX 05	:	SHF/CTR-5
F51D:	84	1161	HEX 84	:	4
F51E:	84	1162	HEX 84	:	CTRL-4
F51F:	44	1163	HEX 44	:	SHIFT-4
F520:	04	1164	HEX 04	:	SHF/CTR-4
F521:	83	1165	HEX 83	:	3
F522:	83	1166	HEX 83	:	CTRL-3
F523:	A3	1167	HEX A3	:	SHIFT-3
F524:	03	1168	HEX 03	:	SHF/CTR-3
F525:	82	1169	HEX 82	:	2
F526:	82	1170	HEX 82	:	CTRL-2
F527:	A2	1171	HEX A2	:	SHIFT-2
F528:	02	1172	HEX 02	:	SHF/CTR-2
F529:	81	1173	HEX 81	:	1
F52A:	81	1174	HEX 81	:	CTRL-1
F52B:	A1	1175	HEX A1	:	SHIFT-1
F52C:	01	1176	HEX 01	:	SHF/CTR-1
F52D:	95	1177	HEX 95	:	(-)
F52E:	95	1178	HEX 95	:	CTRL (-)
F52F:	95	1179	HEX 95	:	SHIFT (-)
F600:	30	1180	HEX 30	:	SHF/CTR (-)
F601:	86	1181	HEX 86	:	6
F602:	86	1182	HEX 86	:	CTRL-6
F603:	A6	1183	HEX A6	:	SHIFT-6
F604:	06	1184	HEX 06	:	SHF/CTR-6
F605:	87	1185	HEX 87	:	7
F606:	45	1186	HEX 45	:	CTRL-7
F607:	A7	1187	HEX A7	:	SHIFT-7
F608:	07	1188	HEX 07	:	SHF/CTR-7
F609:	88	1189	HEX 88	:	8
F60A:	88	1190	HEX 88	:	CTRL-8
F60B:	A8	1191	HEX A8	:	SHIFT-8
F60C:	36	1192	HEX 36	:	SHF/CTR-8
F60D:	89	1193	HEX 89	:	9
F60E:	89	1194	HEX 89	:	CTRL-9
F60F:	A9	1195	HEX A9	:	SHIFT-9
F610:	37	1196	HEX 37	:	SHF/CTR-9
F611:	80	1197	HEX 80	:	0
F612:	80	1198	HEX 80	:	CTRL-0
F613:	AA	1199	HEX AA	:	SHIFT-0
F614:	35	1200	HEX 35	:	SHF/CTR-0
F615:	F1	1201	HEX F1	:	↓
F616:	F1	1202	HEX F1	:	CTRL ↓

F617: F1	1203	HEX F1	: SHIFT +
F618: 3E	1204	HEX 3E	: SHF/CTR ↓
	340	PUT	PANTE2H,02
F619: 09	121	HEX 09	: Y
F61A: 99	12	HEX 99	: CTRL-Y
F61B: 31	13	HEX 31	: SHIFT-Y
F61C: 00	14	HEX 00	: SHF/CTR-Y
F61D: 05	15	HEX 05	: U
F61E: 95	16	HEX 95	: CTRL-U
F61F: 30	17	HEX 30	: SHIFT-U
F620: 0E	18	HEX 0E	: SHF/CTR-U
F621: C9	19	HEX C9	: I
F622: 09	110	HEX 09	: CTRL-I
F623: A0	111	HEX A0	: SHIFT-I
F624: 0F	112	HEX 0F	: SHF/CTR-I
F625: CF	113	HEX CF	: O
F626: 0F	114	HEX 0F	: CTRL-O
F627: 80	115	HEX 80	: SHIFT-O
F628: 43	116	HEX 43	: SHF/CTR-O
F629: 00	117	HEX 00	: P
F62A: 90	118	HEX 90	: CTRL-P
F62B: A8	119	HEX A8	: SHIFT-P
F62C: 44	120	HEX 44	: SHF/CTR-P
F62D: F0	121	HEX F0	: q
F62E: F0	122	HEX F0	: CTRL +
F62F: F0	123	HEX F0	: SHIFT +
F630: 3F	124	HEX 3F	: SHF/CTR +
F631: C8	125	HEX C8	: H
F632: 88	126	HEX 88	: CTRL-H
F633: 21	127	HEX 21	: SHIFT-H
F634: 11	128	HEX 11	: SHF/CTR-H
F635: CA	129	HEX CA	: K
F636: 8A	130	HEX 8A	: CTRL-K
F637: 1F	131	HEX 1F	: SHIFT-K
F638: 1C	132	HEX 1C	: SHF/CTR-K
F639: C9	133	HEX C9	: J
F63A: 88	134	HEX 88	: CTRL-J
F63B: 0E	135	HEX 0E	: SHIFT-J
F63C: 41	136	HEX 41	: SHF/CTR-J
F63D: CC	137	HEX CC	: L
F63E: 8C	138	HEX 8C	: CTRL-L
F63F: C0	139	HEX C0	: SHIFT-L
F640: 42	140	HEX 42	: SHF/CTR-L
F641: 8A	141	HEX 8A	: ;
F642: 8A	142	HEX 8A	: CTRL ;
F643: 88	143	HEX 88	: SHIFT ;
F644: 38	144	HEX 38	: SHF/CTR ;
F645: 80	145	HEX 80	: RETURN
F646: 80	146	HEX 80	: CTRL-RETURN
F647: 80	147	HEX 80	: SHIFT-RETURN
F648: 40	148	HEX 40	: SHF/CTR-RETURN
F649: CE	149	HEX CE	: N
F64A: 8E	150	HEX 8E	: CTRL-N
F64B: 23	151	HEX 23	: SHIFT-N
F64C: 13	152	HEX 13	: SHF/CTR-N
F64D: C0	153	HEX C0	: M
F64E: 80	154	HEX 80	: CTRL-M

F64F: 32	155	HEX 32	: SHIFT-M
F650: 10	156	HEX 10	: SHF/CTR-M
F651: AC	157	HEX AC	: ,
F652: AC	158	HEX AC	: CTRL ,
F653: 0C	159	HEX 0C	: SHIFT ,
F654: 29	160	HEX 29	: SHF/CTR ,
	161		
F655: AC	162	HEX AC	: ,
F656: AC	163	HEX AC	: CTRL ,
F657: BE	164	HEX BE	: SHIFT ,
F658: 3A	165	HEX 3A	: SHF/CTR ,
F659: BF	166	HEX BF	: ?
F65A: BF	167	HEX BF	: CTRL ?
F65B: AF	168	HEX AF	: SHIFT ?
F65C: 26	169	HEX 26	: SHF/CTR ?
	170		
	171	Rotina de interpretacao do buffer IN	
	172		
F65D: A2 00	173	ZXY LOX #500	:Zera X e Y
F65F: A0 00	174	LDY #500	
F661: B9 00 02	175	PROXIN LDA IN,Y	
F664: 05 FD	176	STA RTEMP1	:Armazena caracter
F666: 30 17	177	BRI CACT	:Se e caracter de texto
F668: A0 32 04	178	LDA GRAFFLG	:Se GRAFFLG (<) 0,houve CTRL-B
F66B: 00 24	179	BNE BGRAF	
F66D: A5 FD	180	LDA RTEMP1	
F66F: 20 EA F7	181	JSR SELOFAX	:Seleciona o tipo de caracter
F672: 00 08	182	BCC CACT	:Se caracter grafico
F674: A5 FD	183	LDA RTEMP1	
F676: 0C 0A 0A	184	STY ING+5A	
F679: 20 FE F6	185	JSR PLVMS	
F67C: AC 9C F6	186	JMP STDING	
F67F: A5 FD	187	CACT LDA RTEMP1	
F681: C7 02	188	CMF #502	:E CTRL-B ?
F683: 00 17	189	RNE STDING	
F685: A9 01	190	LDA #501	:Sim, inverte GRAFFLG
F687: A0 32 04	191	ENE GRAFFLG	
F68A: 00 32 04	192	STA GRAFFLG	
F68D: C8	193	INY	
F68E: AC 61 F6	194	JMP PROXIN	
F691: A7 F2	195	BGRAF LDA Bgraf	:Caracter de controle grafico
F693: 90 0E 0A	196	STA ING+5E,X	
F696: EB	197	INX	
F697: A5 FD	198	LDA RTEMP1	
F699: 18	199	CLC	
F69A: 6F 40	1100	ADC #540	:Converte p/ ASC graficos
F69C: 90 0E 0A	1101	STDING STA ING+5E,X	
F69F: C9 00	1102	CMF #580	:E RETURN ?
F6A1: FD 04	1103	BCC ZGRAFFLG	:Sim
F6A3: C8	1104	INY	:Mas
F6A4: EB	1105	INX	
F6A5: 00 0A	1106	BNE PROXIN	:E fim do buffer ?
F6A7: A2 00	1107	ZGRAFFLG LDX #500	
F6A9: 0E 32 04	1108	STX GRAFFLG	
F6AC: 20 30 F6	1109	JSR TRANSBUF	
F6AF: 60	1110	RTS	
	1111		

F680:	03 0E 06	>112	TRANSBUF	LDI	INQ+SE,X	;Reorganiza buffer IN
F683:	90 00 02	>113		STA	IN,X	
F684:	C9 80	>114		CMP	#580	;E RETURN ?
F688:	F0 03	>115		BEQ	FINTRANS	;Sim
F68A:	E8	>116		SNX		;Nao
F68B:	00 F3	>117		BNE	TRANSBUF	;Continua transferindo o buffer
F68D:	60	>118	FINTRANS	RTS		
		>119				
F68E:	20 04 F6	>120	PLVBAS	JSR	EDTBAS	;Calcula em (COOL) o end. na tab.
F6C1:	81 F8	>121	OUTRCH	LDI	(COOL),Y	da palavra BASIC desejada
F6C3:	30 08	>122		BNL	ULTCHR	;Se NULO, entao e ultimo caracter
F6C5:	09 80	>123		ORA	#580	;Transforma em NASCII
F6C7:	90 0E 06	>124		STA	INQ+SE,X	
F6CA:	20 C1 FF	>125		JSR	NXC00	;Proximo caracter da palavra
F6CB:	E8	>126		SNX		;Incrementa INQ
F6CE:	00 F1	>127		BNE	OUTRCH	;Nao e o ultimo caracter ?
F6D3:	AC 06 06	>128	ULTCHR	LDI	INQ+SA	;Ultimo caracter, entao retorna
F6D3:	60	>129		RTS		
		>130				
F6D4:	C9 1E	>131	EDTBAS	CMP	MS1E	;Comp. se o cod. ASCII e (= A NLE
F6D4:	90 02	>132		BCI	SNORS	;Entre M1 e M10 fica como esta
F6D8:	E9 16	>133		SBC	MS16	;Entre M45 e M34 subtrai M16
F6DA:	A8	>134	SNONS	TAY		;Codigo em Y para indexar
F6DB:	80	>135		DEY		;tab.de desloc. de palavras
F6DC:	A0 07 F7	>136		LDI	ENDTBS	;coloca o endereco de inicio
F6DF:	85 F8	>137		STA	COOL	da tab.de palavras BASIC
F6E1:	A0 08 F7	>138		LDI	ENDTBS+1	;SFB=80
F6E4:	85 FC	>139		STA	COOH	;SFC=C3
F6E4:	89 09 F7	>140		LDI	INQPBAS,Y	;Pega desloc.de palavra na tab.
F6E9:	A0 00	>141		LDI	#500	;Zera indexador Y
F6EB:	38	>142		SEC		
F6EC:	E9 80	>143		SBC	#580	
F6EE:	F0 16	>144		BEQ	FENBAS	;Se for = 10 retorna
F6F0:	80 02 06	>145		STA	INQ+52	;armazena o desloc. de palavras
F6F3:	81 F8	>146	CONTR	LDI	(COOL),Y	;Caract.da tab.de palav. BASIC
F6F5:	30 05	>147		BNI	FINPA	;Se ultimo caracter da palavra
F6F7:	20 C1 FF	>148		JSR	NXC00	;incrementa o par (COOL-W)
F6FA:	00 F7	>149		BNE	CONTR	
F6FC:	20 C1 FF	>150	FINPA	JSR	NXC00	;Incrementa o par (COOL-W)
F6FF:	CE 02 06	>151		DEC	INQ+52	;Decrementa contad.de palavras
F702:	F0 02	>152		BEQ	FENBAS	;Retorne se e a palavra
F704:	00 ED	>153		BNE	CONTR	;Nao e a palavra, faz novo loop
F706:	A0	>154	FENBAS	RTS		
F707:	80 C3	>155	ENDTBS	DA	TOKTABL	;End.da tab.de palavras BASIC
		>156				
		>157				*Tabela de contagem de palavras BASIC
		>158				*Fornece o deslocamento dentro da tabela
		>159				
F709:	83	>160	INQPBAS	HEX	83	
F70A:	88	>161		HEX	88	
F70B:	87	>162		HEX	87	
F70C:	86	>163		HEX	86	
F70D:	AC	>164		HEX	AC	
F70E:	BC	>165		HEX	BC	
F70F:	EC	>166		HEX	EC	
F710:	FE	>167		HEX	FE	
F711:	AE	>168		HEX	AE	

F712:	82	1669	HEX	82	
F713:	87	1670	HEX	87	
F714:	C1	1671	HEX	C1	
F715:	C7	1672	HEX	C7	
F716:	B1	1673	HEX	B1	
F717:	84	1674	HEX	84	
F718:	A8	1675	HEX	A8	
F719:	97	1676	HEX	97	
F71A:	80	1677	HEX	80	
F71B:	B2	1678	HEX	B2	
F71C:	C5	1679	HEX	C5	
F71D:	B6	1680	HEX	B6	
F71E:	88	1681	HEX	88	
F71F:	C2	1682	HEX	C2	
F720:	83	1683	HEX	83	
F721:	81	1684	HEX	81	
F722:	A0	1685	HEX	A0	
F723:	C4	1686	HEX	C4	
F724:	93	1687	HEX	93	
F725:	80	1688	HEX	80	
F726:	89	1689	HEX	89	
F727:	8F	1690	HEX	8F	
F728:	C8	1691	HEX	C8	
F729:	85	1692	HEX	85	
F72A:	E2	1693	HEX	E2	
F72B:	A0	1694	HEX	A0	
F72C:	88	1695	HEX	88	
F72D:	89	1696	HEX	89	
F72E:	96	1697	HEX	96	
F72F:	8E	1698	HEX	8E	
F730:	A2	1699	HEX	A2	
F731:	8F	1700	HEX	8F	
F732:	99	1701	HEX	99	
F733:	92	1702	HEX	92	
F734:	91	1703	HEX	91	
F735:	84	1704	HEX	84	
F736:	8A	1705	HEX	8A	
F737:	ED	1706	HEX	ED	
		349	PUT	PARTE2H,02	
		31			
F738:	48	32	BUSTOKH	PMA	;Busca token
F739:	A5 44	33	LDA	AAA	
F73A:	00 04	34	BNE	DECOOTK	
F73B:	A8	35	PLA		
F73C:	4C 50 CA	36	JMP	CONITOK	
F741:	68	37	DECOOTK	PLA	
F742:	84 FD	38	STY	RTENP1	;Salva Y
F744:	A0 2E	39	LDY	MS2E	;Indica ao fim da tabela
F746:	09 09 F7	40	PROXTOK	CMP	INDPRAS,Y
F749:	FD 88	41	BEQ	ETOKEN	;Compara se e o token ?
F74B:	88	42	DEY		;Sim
F74C:	10 F8	43	BPL	PROXTOK	;Proximo
F74E:	A4 FD	44	LDY	RTENP1	;Se nao encontrou
F750:	4C 50 CA	45	JMP	CONITOK	
F753:	C8	46	ETOKEN	JNY	
F754:	98	47	TYA		
F755:	C9 1E	48	CMP	MS1E	

F757: 90 03	119	BCC	CONVTON	ajusta codigo
F759: 18	120	CLC		
F75A: 69 16	121	ADC	#516	
F75C: A4 F0	122	CONVTON	LOF RTENP1	
F75E: 20 89 0E	123	JSR	SEND	imprime caracter
F761: A9 00	124	LOA	#500	
F763: 80 04 04	125	STA	TESTB6	
F766: 4C 1C CA	126	JMP	SNOCIR	
	127			
F769: 84 F9	128	ISCRAS	STY RTENP1	Salva Y
F76B: 8E 09 06	129	STX	IMG+59	Salva X
F76E: 48	130	PWA		Salva codigo ASCII
F76F: A5 46	131	LOA	A46	Verifica contido de A46
F771: F0 0A	132	REQ	IMPBAS	Se = 0
F773: C9 03	133	CMF	#500	
F775: 00 03	134	SME	JPRGC	
F777: 4C 00 90	135	JMP	CART0	Se = 3
F77A: 4C 0C 0B	136	JPRGC	JMP PRGC	Se (),3 SROC
F77D: 68	137	IMPBAS	PLA	Retorna codigo ASCII
F77E: 20 04 F6	138	JSH	EDTBAS	Prep.en (COOL),Y o end.no tab.
F781: B1 F9	139	PALETR	LOA (COOL),Y	da palavra BASIC desejada
F783: 30 0A	140	BNI	ULTVF	Se e MASCII
F785: 09 00	141	ORA	#590	Transforma em MASCII
F787: 20 A9 F1	142	JSR	IDCTP	Identifica CONTROL-P
F78A: 20 C1 FF	143	JSR	MOCD0	Incrementa (COOL)
F78D: 00 F2	144	SME	PALETR	
F78F: 20 A9 F1	145	ULTVF	JSR IDCTP	
F792: AE 09 06	146	LOX	IMG+59	Recupera X
F795: A4 F0	147	LOI	RTENP1	Recupera Y
F797: 60	148	RTS		
	149			
F79B: A5 47	150	PREDET	LOA PGAL	Testa PGAL
F79A: F0 03	151	REQ	VOLTRUF	Se PGAL=0,prep.ret.or.do carac.
F79C: 4C 03 08	152	JMP	PRG3	
F79F: 80 FF 01	153	VOLTRUF	LOA IN-1,X	Pega carac.anterior do buffer
F7A2: 30 38	154	BNI	CCTB	Se e MASCII,verif.se e CTRL-B
F7A4: AB	155	TAT		Se e MASCII salva em Y
F7A5: A5 49	156	LOA	VCTB	Testa VCTB
F7A7: 00 3E	157	SME	JOEGRUF	Decr.buffer e busca novo dado
F7A9: 90	158	TYA		
F7AA: 20 EA F7	159	JSR	SELEFAI	Se VCTB=0 recupera codigo de Y
F7AD: 80 38	160	REQ	JOEGRUF	Se graf.decr.buf e busca dado
F7AF: A5 46	161	LOA	A46	Testa A46
F7B1: F0 0A	162	REQ	VTRFE	Volt.buf.e usa letra da palav.
F7B3: C9 04	163	CMF	#504	
F7B5: F0 03	164	REQ	JPRG12	Se = B4
F7B7: 4C 03 90	165	JMP	CART3	Se () B4
F7BA: 4C 12 0B	166	JPRG12	JMP PRG12	
F7BD: 80 FF 01	167	VTRFE	LOA IN-1,X	Carrega com carac.ant.do buf.
F7C0: 20 04 F6	168	JSR	EDTBAS	Tab. de palav. BASIC em (COOL)
F7C3: 20 82 F9	169	JSR	POSCUR	Prepara cursor
F7C6: A0 00	170	ldy	#500	
F7C8: B1 F9	171	RETCMA	LOA (COOL),Y	Carrega codigo da palav.BASIC
F7CA: 30 06	172	BNI	FINDCC	Se e ultima letra
F7CC: 20 18 FC	173	JSR	DESEDA	Nao,entao retorna o cursor
F7CF: 08	174	INT		Incrementa contador de letras
F7D0: 00 F6	175	SME	RETCMA	Pega proxima letra

F702:	CA	176	FINDCC	DEX		;Decrementa buffer
F703:	20 18 FC	177		JSR DESEEN		;Retorna cursor em posicao
F704:	20 82 FB	178		JSR POSCUR		;Posiciona cursor
F705:	4C 7C FO	179		JMP JCMS		;Pega novo dado no teclado
F70C:	C9 82	180	CCTB	CNF #502		;Compara com CONTROL-B
F70E:	D0 07	181		BNE JOECBUF		;Nao,decr.buf.e busca novo dado
F710:	A9 01	182		LDA #501		;Se e CONTROL-B inverte VCTR
F712:	45 49	183		EOR VCTR		
F714:	85 49	184		STA VCTR		
F715:	CA	185		DEX		;Volta buffer
F717:	4C 10 FO	186	JOECBUF	JMP DECBUF		;Busca novo dado
F71A:	C9 46	187	SELEFAX	CNF #546		;Se caract.=46, seta carry
F71C:	80 09	188		BCC RTCS		(graficos)
F71E:	C9 34	189		CMP #534		;Se (46 e)=34, reseta carry
F720:	80 04	190		BCC RTCR		(palavras basic)
F722:	C9 1E	191		CMP #51E		;Se (#34 e)= #1E, seta carry
F724:	80 01	192		BCC RTCS		(NULL)
F726:	18	193	RTCR	CLC		;Se (#1E, reseta carry (BASIC)
F727:	60	194	RTCS	RTS		
		195				
F72B:	A5 27	196	GRMOULD	LDA GRASH		;GRASH = ASCII EFGH
F72A:	09 10	197		ORA #510		
F72C:	85 27	198		STA GRASH		
F72E:	60	199		RTS		
		1100				
F72F:	FF	1101	HEX	FF		
F800:	4C CA EF	1102	PLOT2	JMP PLOT0		
F803:	4A	1103	PLOTM	LSR		;Se linha impar, seta carry
F804:	00	1104		PRP		;Salva status no stack
F805:	20 F5 FB	1105		JSR A0B		;Acumulador #B result. em delh
F808:	20 73 FC	1106		JSR EPOVID		;Prepara ender.da pag de video
F809:	A5 09	1107		LDA #4H		;Se for coord.mar endereco AS 4
F80A:	20	1108		PLP		;primeiras microlin. de um bloco
F80E:	90 02	1109		BCC UPBLOC		de texto, senao as 4 segundas
F810:	69 0F	1110		ADC #50F		o endereco esta em R4
F812:	85 09	1111	UPBLOC	STA #4H		
F814:	60	1112		RTS		
		1113				
F815:	84 EA	1114	STX	SEA		
F817:	EA	1115	NOP			
F818:	EA	1116	NOP			
F819:	20 00 FB	1117	HLIME	JSR PLOT2		;Plota um bloco LINES
F81C:	C4 2C	1118	HLIME1	CPY #2		;Se chegou na ultima coluna,
F81E:	80 11	1119		BCC RETM4		;Retorna
F820:	C8	1120		INY		;Plota na proxima coluna
F821:	20 C8 EF	1121		JSR PLOT1		
F824:	90 F5	1122		BCC HLIME1		
		1123				
F82A:	69 01	1124	ULIMEZ	ADC #506		
F82B:	4B	1125	ULIME	FMA		
F829:	20 00 FB	1126		JSR PLOT2		;Plota um bloco LINES
F82C:	68	1127		PLA		
F82D:	C5 20	1128		CMP #2		;E ultima linha ?
F82F:	90 F5	1129		BCC ULIMEZ		;Nao,plote novo bloco
F831:	60	1130	RETN4	RTS		;Sim,retorne
		1131				
F832:	A5 2F	1132	CLFSCR	LOF #52F		;Posicao da ultima linha horiz,

FB34:	00 02	1133	BME	CLRSC2	;limpa toda tela
FB36:	A0 27	1134	CLINTOP	LDI #127	;limpa topo da tela (modo MIXED)
FB3B:	04 20	1135	CLRSC2	STY V2	;Última linha a ser apagada em V2
FB3A:	A0 27	1136		LDI #127	
FB3C:	A9 00	1137	CLRSC3	LDA #100	;Seta cor preta 1
FB3E:	05 30	1138		STA COLOR	
FB40:	20 28 FB	1139	JSR	VLDNE	;Plota linhas verticais da coluna
FB43:	88	1140	DEY		;39 ate a coluna 0
FB46:	10 F6	1141	RPL	CLRSC3	
FB46:	60	1142	RTS		
		1143			
FB47:	48	1144	GRASCALC	PHA	;Calcula o endereço base da pag.
FB48:	4A	1145		LSR	;de video referente a linha em
FB49:	29 03	1146		AND #503	baixa resolucao
FB4B:	05 48	1147	ORA	PGH	(da linha 0 a linha 24)
FB4D:	05 27	1148	STA	GRASH	
FB4F:	68	1149		PLA	
FB50:	29 18	1150		AND #518	
FB52:	90 0C	1151	BCC	GRCALC	
FB54:	6F 7F	1152	ADC	#57F	
FB56:	05 26	1153	GRCALC	STA	GRASL
FB5B:	0A	1154		ASL	
FB5F:	0A	1155		ASL	
FB5A:	05 26	1156	ORA	GRASL	
FB5C:	05 26	1157	STA	GRASL	;o endereço esta em GRASL-W
FB5E:	60	1158	RTS		
		1159			
FB5F:	A5 30	1160	NEXTCOL	LDA	COLOR
FB61:	18	1161		CLC	
FB62:	69 03	1162		ADC #503	
FB64:	29 0F	1163	SETCOL	AND #50F	;limita COLOR na faixa de 0-15
FB66:	05 30	1164		STA	COLOR
FB6B:	60	1165	RTS		
		350	PUT	PANTE20,02	
		31			
FB6F:	4A	32	SCRN	LSR	;Testa linha par ou impar
FB6A:	08	33		PHP	
FB6B:	20 47 FB	34	JSR	GRASCALC	;Calc.ende.base da pag de video
FB6E:	28	35		PLP	
FB6F:	F0 03	36	BCC	PGHCC	;Se e linha par
FB71:	20 FB F7	37	JSR	GRSHOULD	;Japar, endereco o bloco 2
FB74:	81 26	38	PERCC	LDA (GRASL),Y	;A = Byte cor
FB76:	40	39		PHA	;Salva A no stack
FB77:	98	310		TYA	;Testa se e coluna impar ou par
FB78:	4A	311		LSR	
FB79:	68	312		PLA	;Recupera A do stack
FB7A:	90 03	313	BCC	POCC	;Se coluna par, cor em POCC
FB7C:	20 28 EF	314	JSR	BUSCOR	;Se impar, cor em BUSCOR
FB7F:	4A	315	POCC	LSR	
FB80:	38	316		SEC	
FB81:	4C 45 FA	317	JMP	SCRNG	;Numero da cor no das do acumul.
FB84:	EA	318		NOP	
FB85:	A6 3A	319	INSDS1	LDI	PCL
FB87:	A4 38	320		LDI	PCH
FB89:	20 96 FD	321	JSR	PRTX2	;Imprime PC como ende. de hex.
FB8C:	20 40 F9	322	JSR	PRFLNE	;Imprime 3 espacos em branco
FB8F:	A1 3A	323	INSDS2	LDA	(PCL,X)
					;Busca opcode

FB91:	A6	124	INSDI3	TAY		Guarda em T
FB92:	AA	125		LSR		Testa par ou impar
FB93:	90 0F	126		BCC	IEVEN	Se par IEVEN
FB95:	AA	127		ROR		Testa bit 1
FB96:	80 10	128		BCS	ERR	XXXXXXXX de invalido
FB98:	C9 A2	129		CMF	MSA2	
FB9A:	F0 0C	130		RE4	ERR	Decode 589 invalido
FB9C:	29 87	131		AND	MS87	Mascara bits
FB9E:	4A	132	IEVEN	LSR		LSR para carry para teste
FB9F:	AA	133		TAX		
FBAD:	80 62 F9	134		LDA	FMT1,X	Busca byte de index. de formato
FBAD:	20 65 FA	135		JSR	SCRM2	Se carry=1 habilita DNS no des,
		136	*			Se nao habilita so des
FBAA:	00 04	137		RMC	GETFMT	Se nao erro GETFORM
FBAB:	A0 80	138	ERR	LOI	MS80	Substitui 800 por ce invalido
FBAA:	AF 00	139		LDA	MS00	Indexa FMT2 com zero
FBAC:	AA	140	GETFMT	TAX		
FBAD:	80 A6 F9	141		LDA	FMT2,X	Indexacao para tabela de
FBAD:	85 2E	142		STA	FORMAT	formato de mensagens
FB82:	29 03	143		AND	MS03	Mascara 2 bits de comecimento
FB84:	85 2F	144		STA	LEMETH	(0=10byte, 1=20bytes, 2=30bytes)
FB86:	98	145		TYA		Recupera caractere
FB87:	29 8F	146		AND	MS8F	
FB89:	AA	147		TAX		Opcode masc. por 10001010, em X
FB8A:	98	148		TYA		Decode
FB8B:	A0 00	149		LDY	MS00	
FB8D:	ED 8A	150		CPX	MS8A	
FB8F:	F0 08	151		RE4	MHNDI3	
FB81:	4A	152	MHNDI1	LSR		
FB82:	90 08	153		BCC	MHNDI3	Forma indexacao para
FB84:	4A	154		LSR		tabela de anemonicas
FB85:	4A	155	MHNDI2	LSR		(1) XXXX010 => 0010XXXX
FB86:	09 20	156		ORA	MS20	(2) XXXYY01 => 0011XXXX
FB88:	88	157		DEY		(3) XXXYY10 => 0011XXXX
FB89:	00 FA	158		RMC	MHNDI2	(4) XXXYY100 => 0010XXXX
FB8B:	C8	159		JNY		(5) XXXX000 => 0001XXXX
FB8C:	80	160	MHNDI3	DEY		
FB8D:	00 F2	161		RMC	MHNDI1	
FB8F:	60	162		RTS		
FB8D:	20 85 F8	163	INSTOSP	JSR	INSDI1	Formato e comecimento instricao
FB83:	48	164		PHA		Salva index da tab.de anemonicas
FB84:	81 3A	165	PRN*CP	LDA	(FCL),Y	Carrega Byte
FB86:	20 04 F0	166		JSR	PRBYTE	
FB89:	A2 01	167		LDX	MS01	Imprime 1 espaco
FB8B:	20 AA F9	168	PRINTBL	JSR	PRBL2	
FB8C:	C4 2F	169		CPY	LEMETH	Compara quantos bytes devem ser
FB8D:	C8	170		JNY		impressos (de 1 a 3)
FB81:	90 F1	171		BCC	PRNTOP	Mod imprimatodos" antes o prog.
FB83:	A2 03	172		LDX	MS03	Tabela usado para imprimanem.
FB85:	00 04	173		CPY	MS04	
FB87:	90 F2	174		BCC	PRINTBL	
FB89:	68	175		PLA		Recupera index. de anem. na T
FB8A:	A8	176		TAY		
FB8B:	89 C0 F9	177		LDA	MHNDL,Y	Busca 3 caract.anem. em 2 Bytes
FB8E:	85 2C	178		STA	LMHEN	
FB8D:	89 00 FA	179		LDA	MHENDR,Y	
FB83:	95 20	180		STA	RMHEN	

F8F5: A7 00	>81	PRM1	LDI	MS00	
F8F7: A0 05	>82		LDI	MS05	
F8F9: 06 20	>83	PRM2	ASL	R404H	
F8FB: 26 2C	>84		ROL	L404H	
F8FD: 2A	>85		ROL		;Desloca 5 bits do caracter
F8FF: 08	>86		DEY		;no acum. (limpa carry)
F901: 00 FB	>87	BNE	PRM2		
F903: 6F 8F	>88	ADC	#??		;Converte ASCII (usua OFFSET??)
F905: 20 ED FD	>89	JSR	COU7		;Imprime caractere (letra) ou (?)
F90A: 0A	>90	DEX			
F90D: 00 EC	>91	BNE	PRM1		;Imprime ate a terceira letra
F90F: 20 4B F9	>92	JSR	PRBLNK		;Imprime 3 espacos em branco
F911: 64 2F	>93		LDI	LENGTH	
F913: A2 06	>94		LDI	MS06	;Indica CHAR1 e CHAR2
F915: ED 03	>95	PRADR1	CPX	MS03	
F917: FD 1C	>96	BEQ	PRADR5		;Se X=3 retaa, vai para PRADR5
F91A: 06 2E	>97	PRADR2	ASL	FORMAT	;Prepara indexacao
F91C: 90 0E	>98	BCC	PRADR3		;Imprime os form.(S,M,I,...,X,Y)
F91E: 00 83 F9	>99	LDI	CHAR1-1,X		
F920: 20 6B FB	>100	JSR	COU7		
F922: 80 8F F9	>101	LDI	CHAR2-1,X		
F924: FD 03	>102	BEQ	PRADR3		
F926: 20 ED FD	>103	JSR	COU7		
F928: 0A	>104	PRADR3	DEX		
F92A: 03 E7	>105	BNE	PRADR1		
F92C: 60	>106	RTS			
	>107				
F92E: 88	>108	PRADR4	DEY		
F930: 20 E7	>109	BNE	PRADR2		
F932: 20 0A FD	>110	JSR	PRBYTE		;Impr. contido no ender. do acum.
F934: AC 2E	>111	PRADR5	LDI	FORMAT	
F936: C9 EB	>112	CMP	MS09		
F938: 81 3A	>113	LDI	PCU1,Y		
F93A: 90 F2	>114	BCC	PRADR4		
F93C: 20 5A F9	>115	RELAOR	JSR	PCADJ3	
F93E: 0A	>116	TAX			;PCU1,PCU=(OFFSET+1) PARA A,Y
F940: 65	>117	INX			
F942: 00 01	>118	BNE	PRMTYX		;+1 PARA X,X
F944: C8	>119	INP			
F946: 92	>120	PRMTYX	TYA		
F948: 20 0A FD	>121	PRMTAX	JSR	PRBYTE	;Imprime o acumulador em HEX
F94A: 0A	>122	PRMTX	TXA		
F94C: AC 06 FD	>123	JMP	PRBYTE		;Imprime o acumulador em HEX
F94E: A2 03	>124	PRBLNK	LDI	MS03	;Prepara X para contar 3 vezes
F950: A9 40	>125	PRBL2	LDI	#"	;Carrega o acumulador com "espaco"
F952: 20 ED FD	>126	PRBL3	JSR	COU7	;Imprime espaco "X" vezes
F954: 0A	>127	DEX			
F956: 00 FB	>128	BNE	PRBL2		
F958: 40	>129	RTS			
	>130				
F95B: 38	>131	PCADJ1	SEC		;LENGTH -> 0=1-byte , 1=2-bytes
F95D: A5 2F	>132	PCADJ2	LDI	LENGTH	2=3-bytes
F95F: 6A 3B	>133	PCADJ3	LDI	PCN	
F961: 0A	>134	TAX			
F963: 10 01	>135	BPL	PCADJ4		;Testa sinal do displacement
F965: 08	>136	DEY			

F95C:	45 3A	>137	PCADJ4	ADC	PCL	:PCL + LENGTH (ou displacement) +1
F95E:	90 08	>138		RCC	RETMS	para A carry em Y (PCN)
F960:	CB	>139			INY	
F962:	60	>140	RETMS	RTS		
		351		POT	PARTE2P,02	
		31				
		32			* INTES FMT1 XXXXXX10 IMSTRS	
		33			* Se b=0, estao meio byte a esquerda	
		34			* Se Y=1, estao meio byte a direita	
		35			* (X=IMOE4)	
		36				
F962:	04 20 54	37	FMT1	HEX	042054	
F965:	30 00	38		HEX	3000	
F967:	80 04 90	39		HEX	800490	
F96A:	03 22	310		HEX	0322	
F96C:	54 33 00	311		HEX	543300	
F96F:	80 04	312		HEX	8004	
F971:	90 04 20	313		HEX	900420	
F974:	54 33	314		HEX	5433	
F976:	00 80 04	315		HEX	008004	
F979:	90 04	316		HEX	9004	
F97B:	20 54 30	317		HEX	205430	
F97E:	03 80	318		HEX	0380	
F980:	04 90 00	319		HEX	049000	
F983:	22 44	320		HEX	2244	
F985:	33 00 08	321		HEX	330008	
F988:	44 00	322		HEX	4400	
F98A:	11 22 44	323		HEX	112244	
F98C:	33 00	324		HEX	3300	
F98F:	08 44 A9	325		HEX	0844A9	
F992:	01 22	326		HEX	0122	
F994:	44 33 00	327		HEX	443300	
F997:	80 04	328		HEX	8004	
F999:	90 01 22	329		HEX	900122	
F99C:	44 33	330		HEX	4433	
F99E:	00 90 04	331		HEX	009004	
F9A1:	90	332		HEX	90	
F9A2:	26 31 87	333		HEX	263187	
F9A5:	9A	334		HEX	9A	
F9A6:	00	335	FMT2	DFB	500	ENRO
F9A7:	21	336		DFB	521	IMEDIATO
F9A9:	81	337		DFB	581	Z-PAGE
F9AB:	82	338		DFB	582	ABSOLUTO
F9AD:	00	339		DFB	500	AMPLIADO
F9AE:	00	340		DFB	530	ACUMULADOR
F9AC:	59	341		DFB	559	(ZPAG,X)
F9AD:	40	342		DFB	540	(ZPAG),Y
F9AE:	91	343		DFB	591	ZPAGE,X
F9AF:	92	344		DFB	592	ABS,X
F9B0:	8A	345		DFB	58A	ABS,Y
F9B1:	4A	346		DFB	54A	(ABS)
F9B2:	85	347		DFB	585	ZPAG,Y
F9B3:	90	348		DFB	590	RELATIVO
		349				
F9B4:	AC	350	CHM1	ASC	"."	
F9B5:	AF	351		ASC	"1"	
F9B6:	AC	352		ASC	"."	

F9B7:	A3	153	ASC	"H"	
F9B8:	A8	154	ASC	"I"	
F9B9:	A4	155	ASC	"J"	
		156			
F9BA:	09	157	CHAR2	ASC	"Y"
F9BB:	00	158	HEX	00	
F9BC:	08	159	ASC	"X"	
F9BD:	A4	160	ASC	"J"	
F9BE:	A4	161	ASC	"J"	
F9BF:	00	162	HEX	00	
		163			
		164	* HHEML e da forma:		
		165	*	(A)	XXXXX000
		166	*	(B)	XXXXY000
		167	*	(C)	XXXXX010
		168	*	(D)	XXXXYY10
		169	*	(E)	XXXXYY00
		170	*		(X=INDEX)
		171			
F9C0:	1C BA 1C	172	HHEML	HEX	1CBA1C
F9C3:	23 5D 88	173		HEX	235D88
F9C6:	18 A1 99	174		HEX	18A199
F9C9:	0A 1D 23	175		HEX	0A1D23
F9CC:	9D 08 18	176		HEX	9D0818
F9CF:	A1 0D 29	177		HEX	A10D29
F9D2:	19 AE 69	178		HEX	19AE69
F9D5:	A8 19 23	179		HEX	A81923
F9D8:	24 53 18	180		HEX	245318
F9DB:	23 24 53	181		HEX	232453
F9DE:	19 A1	182		HEX	19A1
F9E0:	00 1A 5B	183		HEX	001A5B
F9E3:	58 A5 69	184		HEX	58A569
F9E6:	24 24	185		HEX	2424
F9E9:	AE AE A8	186		HEX	AEAEA8
F9EC:	A0 29 0D	187		HEX	A0290D
F9EF:	7C 0D	188		HEX	7C0D
F9F0:	15 9C 6D	189		HEX	159C6D
F9F3:	9C A5 69	190		HEX	9CA569
F9F6:	29 53	191		HEX	2953
F9F9:	04 13 34	192		HEX	041334
F9FB:	11 A5 69	193		HEX	11A569
F9FE:	23 A0	194		HEX	23A0
FA00:	08 62 5A	195	HHEML	HEX	08625A
FA03:	49 26 62	196		HEX	492662
FA06:	14 09 54	197		HEX	140954
FA09:	44 C9 54	198		HEX	44C954
FA0C:	68 44 E8	199		HEX	6844E8
FA0F:	14 08 84	200		HEX	140884
FA12:	08 04 74	201		HEX	080474
FA15:	84 28 6E	202		HEX	84286E
FA18:	74 74 0C	203		HEX	74740C
FA1B:	4A 72 F2	204		HEX	4A72F2
FA1E:	A4 BA	205		HEX	A4BA
FA20:	0D A4 A2	206		HEX	0DA4A2
FA23:	A2 74 74	207		HEX	A27474
FA26:	74 72	208		HEX	7472
FA29:	44 68 82	209		HEX	446882

FA29: 32 B2 00	1110	HEX 32B200	
FA2E: 22 00	1111	HEX 2200	
FA30: 1A 1A 26	1112	HEX 1A1A26	
FA33: 26 72 72	1113	HEX 267272	
FA36: 08 C8	1114	HEX 88C8	
FA38: 04 CA 26	1115	HEX C4CA26	
FA3B: 48 44 44	1116	HEX 484444	
FA3E: A2 C8	1117	HEX A2C8	
FA40: 85 45	1118	DBB STA ACC	;Salva acumulador em ACC
FA42: 68	1119	PLA	
FA43: 48	1120	PHA	
FA44: 04	1121	ASL	
FA45: 04	1122	ASL	
FA46: 04	1123	ASL	
FA47: 30 03	1124	BNZ BREAK	;Teste para BREAK
FA49: AC FE 03	1125	JMP (196LOC)	
	1126		
FA4C: 28	1127	BREAK P/LP	
FA4D: 20 9F F8	1128	JSR SAV2	;Salva regs.no BREAK, incluindo PC
FA50: 68	1129	FLA	
FA51: 85 3A	1130	STA PCL	
FA53: 68	1131	PLA	
FA54: 85 38	1132	STA PCH	
FA56: AC FD 03	1133	JMP (889UL)	;Normalmente JMP XXXX
FA59: 28 85 F8	1134	JSR INSDS1	;Imprime PC no BREAK
FA5C: 20 CF FA	1135	JSR RGDSPL	;Imprime conteúdo dos regs.
FA5F: 4C 50 FF	1136	JMP MON	;Retorna no início do monitor
	1137		
	1138		
	1139		***No ligar o aparelho o vetor RESET indica este endereço
FA62: 4C AE FA	1140	RESET JMP RESET1	
	1141		
FA65: 90 04	1142	SCBH2 BCC BTHSKZ	
FA67: 4A	1143	LSR	;Transfere o DMS para o des
FA68: 4A	1144	LSR	
FA69: 4A	1145	LSR	
FA6A: 4A	1146	LSR	
FA6B: 29 0F	1147	BTHSKZ AND #50F	;Zera o DMS
FA6D: 40	1148	RTS	
	1149		
FA6E: 08	1150	RESET1 CLD	;Limpa modo decimal
FA6F: 20 94 FE	1151	JSR SETMODH	;Video na forma normal
FA72: 28 38 F8	1152	JSR INI71	;Inicia memoria e formatos
FA75: 20 93 FE	1153	JSR SETVIO	;Video como disp.de saída de dados
FA78: 20 99 FE	1154	JSR SETKBO	;Teclado como entrada de dados
FA7B: 20 3A FF	1155	JSR BELL	;Gera um BELL
FA7E: A0 F3 03	1156	LDA SOFTEWH	;Verif. se e warm ou cold start
FA81: 49 A5	1157	EOR #5A5	
FA83: C0 F4 03	1158	CMF #A8EDUP	
FA86: D0 10	1159	BNE PPREG	;Se diferente faz um cold start
FA8B: A0 F2 03	1160	LDA SOFTEVL	;Faz um warm start
FA8B: D0 12	1161	BNE EXT10	;Se warm start vai para EXT10
FA8B: A9 C2	1162	LDA #5C2	
FA8F: C0 F3 03	1163	CMF SOFTEWH	
FA92: D0 08	1164	BNE EXT10	
FA94: A0 03	1165	FIXSEV LDT #5D3	;SOFTEVL= #5D3

```

FAR6: 8C F2 03 1166 STY SOFTEVL
FAR7: 20 33 EF 1167 JSR EXTIO1 ;Verifica presenca de perifericos
FAR8: 4C 00 C2 1168 JMP BASDC ;Inicio do BASIC
FARF: 20 33 EF 1169 EXTIO JSR EXTIO1 ;Verifica presenca de perifericos
FAG2: 6C F2 03 1170 WOFIX JMP (SOFTEVL)
FAG3: 20 79 EC 1171 PPREG JSR PPREG
FAG8: 20 5F FB 1172 PARUP JSR OPTV2000 ;Limpa tela, imprime TK2000
FAGB: A2 05 1173 LDX #005 ;Transfere a tabela VETSTART
FAGD: 80 97 FB 1174 SETPLP LDA VETSTRT-1,X para os vetores
FAGF: 90 EF 03 1175 STA @RVL-1,X (@RVL, (SOFTEV), PARUPL
FAG8: CA 1176 OEX
FAG4: 00 F7 1177 BNE SETPLP ;Carrega ate o ultimo vetor
FAG6: 86 00 1178 STX LOCD ;(LOCD) = $C180
FAG8: A9 C1 1179 LDA #C1
FAGA: 85 01 1180 STA LOC1
FAGC: A0 07 1181 SLOOP LOT #D07 ;Y = 87
FAG8: 81 00 1182 NXTBYT LDA (LOCD),Y
FAGD: 09 BF F5 1183 CMP DISKID-1,Y ;Compara (LOCD), Y com DISKID, Y
FAGF: 20 CF 1184 BNE FIXEVE ;Se for 1) va para FIXEVE
FAG8: 88 1185 DEY
FAGC: 88 1186 DEY
FAG7: 1D F5 1187 BPL NXTBYT ;Compara ate ultimo byte
FAG9: 4C 00 00 1188 JMP (LOCD) ;Jump para diskdirver
1189
352 PUT PARTE2P-1,02
FAGC: 20 BE F0 11 REGOSP JSR CROUT ;Faz um RETURN
FAGF: A9 F0 12 REGOSP LDA #AREG ;A3= $07F0 (AREG)
FAG8: 85 40 13 STA A3L
FAGD: A9 07 14 LDA #NBEG
FAGF: 85 46 15 STA A3H
FAG7: A2 F8 16 LDX #5FB ;X como contador
FAG9: A9 A0 17 RDSP1 LDA #* * ;Imprime um espaco
FAGB: 20 ED F0 18 JSR COUNT
FAGD: 80 CA F0 19 LDA RTBL-251,X ;Imprime o nome do registrador
FAGF: 20 ED F0 110 JSR COUNT
FAG8: A9 80 111 LDA #"" ;Imprime o caracter = (igual)
FAGC: 20 ED F0 112 JSR COUNT
FAGF: 80 F5 06 113 LDA AREG-251,X ;Impr.o contedo do reg. em hex
FAGC: 20 DA F0 114 JSR PRINTTE
FAGF: FB 115 DMX
FAGD: 30 E7 116 BNE RDSP1 ;Represente todos registradores
FAGF: 60 117 RTS
118
FAG3: 89 F3 F0 119 NENS LDA #ENTAB,Y ;Imprime caracter de ENTAB,Y
FAG6: 20 ED F0 120 JSR COUNT
FAGF: C8 121 DMY ;Proximo caracter
FAGA: CA 122 DEX
FAG8: 80 F6 123 BNE NENS ;Ate o ultimo
FAGD: 60 124 RTS
125
FAGC: FF FF FF 126 HEX FFFFFFFF
FAG8: FF
FAG2: 20 14 F0 127 IN2 JSR IN3
FAG5: A5 27 128 SEPTAB LDA #BASH ;Separa a tabela de codigos ASCII
FAG7: 0A 129 ASL ;de 4 em 4
FAG8: 0A 130 ASL
FAG9: 24 27 131 BIT #BASH

```

FB08:	08	132		PHP		
FB0C:	49 00	133		ADC #500	;Se CTRL soma 1 ao acum. (carry)	
FB0E:	28	134		PLP		
FB0F:	10 03	135		RPL HTCTL	;Se SHIFT/CTRL,+3 ao acum.(2+carry)	
FB11:	18	136		CLC	;Se e SHIFT soma 2 ao acum.	
FB12:	49 02	137		ADC #502		
FB14:	0E 0E 04	138	HTCTL	STX XSAV	;Salva X em XSAV	
FB17:	AA	139		TAX	;Prepara indexacao	
FB19:	80 9D F5	140		LDA TABASC,X		
FB1B:	AE 0E 04	141		LDR XSAV	;Recupera X em XSAV	
FB1E:	38	142		SEC		
FB1F:	60	143		RTS		
		144				
FB2B:	AA 24	145	KETPOS	LDY CH	;?n posicao horizontal do cursor	
FB2D:	20 80 F0	146		JSR PRCV01	;Imprime caracter no video	
FB2F:	E6 24	147	KETPOS1	INC CH	;Desloca cursor a direita	
FB2F:	A5 24	148		LDA CH	;Ultrapassou a largura da tela ?	
FB2F:	C5 21	149		CMP WNDMOTN	;Sim, faz um CODCR	
FB2F:	80 01	150		BCS JCRCR		
FB2F:	60	151		RTS		
		152				
FB2E:	4C 17 F1	153	JCRCR	JMP CODCR		
FB31:	A4 20	154	POSESB	LDY WNDLFT	;Posiciona cursor a esquerda	
FB33:	04 24	155		STY CH		
FB35:	60	156		RTS		
		157				
FB36:	FF FF FF	158		HEX FFFFFFFF		
FB39:	FF FF					
FB3B:	20 79 EC	159	INIT1	JSR PRPREO		
FB3E:	A9 00	160		LDA #500	;Zera status	
FB40:	80 F3 07	161		STA STATUS		
FB43:	A9 00	162		LDA #500	;Faz topo do texto na linha 0	
FB45:	F0 05	163		BEQ SETWNO		
FB47:	20 36 F8	164		JSR CLRTOP	;limpa parte superior da tela	
FB4A:	A9 14	165		LDA #514	;Faz topo do texto na linha #514	
FB4C:	85 22	166	SETWNO	STA WNDTOP		
FB4E:	A9 00	167		LDA #500	;0,como inicio a esquerda da tela	
FB50:	85 20	168		STA WNDLFT		
FB52:	A9 28	169		LDA #528	;Prepara largura da tela com #528	
FB54:	85 21	170		STA WNDMOTN		
FB56:	A9 18	171		LDA #518	;Prepara a #518 como ultima linha	
FB58:	85 23	172		STA WNDOTN		
FB5A:	A9 17	173		LDA #517	;Cursor vertical na ultima linha	
FB5C:	85 25	174	TABV	STA CV		
FB5E:	60	175		RTS		
		176				
FB5F:	20 58 FC	177	DPK2000	JSR HOME	;Faz um HOME	
FB62:	A9 0E	178		LDA #50E	;Posiciona cursor horizontal	
FB64:	85 24	179		STA CH	;Na coluna #50E	
FB66:	A0 06	180		LDY #506		
FB68:	89 96 F5	181	STITLE	LDA TITULO-1,Y	;Pega caracter de titulo	
FB6B:	20 80 F0	182		JSR PRCV01	;Imprime o caracter	
FB6E:	E6 24	183		INC CH	;Proxima coluna	
FB70:	88	184		DEY		
FB71:	00 F5	185		BNE STITLE	;Até o ultimo caracter	
FB73:	60	186		RTS		
		187				

FB74: 40 F3 03	188	LDA	SOFTSWH	
FB77: 4F A5	189	EOR	MSAS	
FB79: 80 F4 03	190	STA	PMEDUP	
FB7C: 60	191	BTS		
	192			
FB7D: C9 80	193	VDMAIT	CMP	MSB0 ;Compara se e um RETURN
FB7F: 00 14	194	BNE	MSMAIT	
FB81: 20 43 F0	195	JSR	SCAH1	;Se e RETURN varre teclado
FB84: 90 00	196	BCC	LDRET	;Se nao tem tecla apertada
FB86: C9 93	197	CMP	MSF3	;E CONTROL-S ?
FB8B: 00 09	198	BNE	LDRET	
FB8A: 20 43 F0	199	JPSCM1	JSR	SCAH1 ;Tem tecla apertada ?
FB8D: 90 F8	100	BCC	JPSCM1	;loop ate ter tecla apertada
FB8F: C9 91	101	CMP	MSF1	;E CONTROL-B
FB91: 00 F7	102	BNE	JPSCM1	;Se nao, novo loop
FB93: A9 80	103	LDRET	LDA	MSB0 ;Prepara para um RETURN
FB95: 4C F0 F8	104	MDMAIT	JMP	IOFC00
	105			
FB98: 5F	106	VTSTRT	HEX	5F
FB99: FA	107		HEX	FA
FB9A: 00	108		HEX	00
FB9B: C2	109		HEX	C2
FB9C: 67	110		HEX	67
	111			
FB9D: 85 45	112	SAVE1	STA	ACC ;Salva o acumulador em ACC
FB9F: 8E F1 07	113	SAVE2	STX	XREG ;Salva X em XREG
FBA2: A5 45	114	LDA	ACC	;Salva o acumulador em AREG
FBA4: 80 F0 07	115	STA	AREG	
FBA7: 4C 40 FF	116	JMP	SAV3	;Salva Y, status e stack pointer
	117			
FBA4: 00 2A 55	118	TABCOR	HEX	'002A557F50A05FF
FBA7: 7F 80 A0 05	119			
	119			
FB82: 48	1120	POSCUR	PNA	;Salva acumulador e Y no stack
FB83: 98	1121		TYA	
FB84: 48	1122		PNA	
FB85: 20 F3 F8	1123	JSR	CVB8	;DEL= CVB8
FB88: 20 73 FC	1124	JSR	EPB10	;Prep.B4 com end. da pag de video
FB8E: A0 00	1125	L0F	MSB0	;Zera Y (indicador)
FB90: 18	1126	CLC		;Sltiza posic.horiz.na pag de vid.
FB8E: A5 00	1127	LOA	R4L	
FBC0: 45 24	1128	ADC	CX	
FBC2: 95 08	1129	STA	R4L	
FBC4: 81 08	1130	IMACUR	LDA	(R4L),Y ;prepara cursor como caracter de
FBC6: 4F 7F	1131	EOR	MS7F	8 microlinhas no modo inverso
FBC8: 91 08	1132	STA	(R4L),Y	
FBCA: A5 09	1133	LDA	R4H	;Proxima microlinha
FBCC: 6F 04	1134	ADC	MS04	
FBCD: 85 09	1135	STA	R4H	
FBD0: 29 40	1136	AND	MS40	
FBD2: F0 F0	1137	BEQ	IMACUR	;pste a ultima microlinha
FBD4: A8	1138	PLA		;Recupera Y e acumulador do stack
FBD5: A8	1139	TAY		
FBD6: 68	1140	PLA		
FBD7: 60	1141	BTS		
	353	P07	PARTE26_02	
	31			

FBD6: EA	12		NOP		
FBD9: C9 07	13	BELL0	CMF	Bell1	;Compara se o caracter BELL
FBD1: D3 12	14		BNE	RTS28	;Se nao for retorna
FBD4: A9 40	15	BELL1	LDA	MS40	
FBD5: 20 A8 FC 16			JSR	WAIT	;Atraso de .022 segundos
FBE2: A0 C0	17		LDR	MS00	;Gera um som a 1MHz
FBE4: A9 DC	18	BELL2	LDA	MS0C	de duracao de .18953 segundos
FBE6: 20 A8 FC 19			JSR	WAIT	
FBE9: A0 30 C0 110			LDA	SPKR	
FBE0: 00	111		BEY		
FBE0: D0 F5	112		BNE	BELL2	
FBEF: 40	113	RTS28	RTS		
	114				
FBF0: 4C 20 F8	115	NOPOS1	JMP	NOPOS6	;imprime caracter e avanca cursor
FBF3: A5 25	116	CVXB	LDA	CV	;Multiplica CV x 8
FBF5: 0A	117	AND	ASL		;Armazena resultado em DELH
FBF6: 0A	118		ASL		;Obs: CV continua intacto
FBF7: 0A	119		ASL		
FBF8: 85 E3	120		STA	DELH	
FBF9: 40	121		RTS		
	122				
FBF0: EA	123		NOP		
FBF0: EA	124		NOP		
FBF0: C9 F0	125	INFC00	CMF	MSF0	
FBF7: F0 22	126		BEQ	UP	;Suba uma linha
FC01: C9 F1	127		CMF	MSF1	
FC03: F0 41	128		BEQ	LF	;Desce uma linha
FC05: C9 A0	129	VIDOUT	CMF	MS40	;Se for != A0 ou != 7F
FC07: 80 E7	130		BCC	NOPOS1	ir para NOPOS1
FC09: A8	131		TAT		
FC0A: 10 E4	132		RPL	NOPOS1	
FC0C: C9 80	133		CMF	MS80	;Se for RETURN faz um CR
FC0E: F0 52	134		BEQ	CR	
FC10: C9 8A	135		CMF	MS8A	;Se for N8A (CTRL-K) faz um LF
FC12: F0 52	136		BEQ	LF	
FC14: C9 88	137		CMF	MS88	;E MS88 (CONTROL-H) ?
FC16: D0 C1	138		BNE	BELL0	;Se nao, BELL0
FC18: A5 24	139	DESE04	LDA	CR	;Se o cursor nao estiver
FC1A: C5 20	140		CMF	MS0LFT	totalmente a esquerda da tela,
FC1C: D0 02	141		BNE	DESE04	desloque uma posicao a esquerda
FC1E: E8	142		INX		
FC1F: 60	143		RTS		
FC20: C6 24	144	DESE04	DEC	CR	
FC22: 60	145		RTS		
	146				
FC23: A5 22	147	UP	LDA	MS0TOP	;Se o cursor nao esta no topo
FC25: C5 25	148		CMF	CV	sob o uma linha
FC27: 80 00	149		BCC	RTS4	
FC29: C6 25	150		DEC	CV	
FC2B: EA	151	RTS4	NOP		
FC2C: 60	152	RTS4	RTS		
	153				
FC20: A9 DC	154	PMES4	LDA	MS0C	;imprime o caracter "\n" (CTRL-X)
FC2F: 20 E0 F0	155		JSR	COUT	
FC32: 60	156		RTS		
	157				
FC33: 20 18 F0	158	IM5	JSR	IM4	;Testa se tem tecla apertada

FC96: 85 08	1116	STA	R4L	
FC98: 60	1117	RTS		
	1118			
FC99: A4 24	1119	CLREL	L0Y	CH
FC9B: A9 A0	1120	CLRELZ	L0N	0" "
FC9D: 20 86 FD	1121	CLEVEP	JSR	PROVID
FC9B: C8	1122	IMY		
FC9C: CA 21	1123	CPY	WINDMTH	
FC9D: 9D FB	1124	BCC	CLEREK	
FC9E: A4 20	1125	L0Y	WINDLFT	
FC97: 60	1126	RTS		
	1127			
FC9B: 38	1128	WAIT	SEC	
FC99: 48	1129	WAIT2	PNA	
FC9A: E9 D1	1130	WAIT3	SBC	WS01
FC9C: 00 FC	1131	BNE	WAIT3	
FC9E: 68	1132	PLA		
FC9F: E9 D1	1133	SBC	WS01	
FCB1: 00 F6	1134	BNE	WAIT2	
FCB3: 60	1135	RTS		
	1136			
FCB4: E6 42	1137	NXTA4	INC	A4L
FCB6: 00 02	1138	BNE	NEXTA1	
FCB8: E6 43	1139	INC	A4H	
FCBA: A5 3C	1140	NEXTA1	L0N	ASL
FCBC: C5 3E	1141	CMY	A2L	
FCBE: A5 30	1142	L0N	A1H	
FCB8: E5 3F	1143	SBC	A2H	
FCCE: E6 3C	1144	INC	A1L	
FCCE: 00 02	1145	BNE	RTS48	
FCCE: E6 30	1146	INC	A1H	
FCCE: 60	1147	RTS48	RTS	
	1148	PUT	PARTE2R,02	
	1149			
FCCE: A0 48	1150	HEADN	L0Y	WS48
FCCE: 20 08 FC	1151	JSR	ZERDLY	
FCCE: 00 F9	1152	BNE	HEADR	
FCCE: 69 FE	1153	ADC	WSFE	
FCCE: 80 F5	1154	BCC	HEADR	
FCCE: A0 21	1155	L0Y	WS21	
FCCE: 20 08 FC	1156	JSR	ZERDLY	
FCCE: C8	1157	INT		
FCCE: C8	1158	INT		
FCCE: 88	1159	ZERDLY	DEY	
FCCE: 00 FD	1160	BNE	ZERDLY	
FCCE: 90 05	1161	BCC	WRITAP	
FCCE: A0 32	1162	L0Y	WS32	
FCCE: 00	1163	WRITAP	DEY	
FCCE: 00 FD	1164	BNE	WRITAP	
FCCE: AC 20 C0	1165	L0Y	TAPEOUT	
FCCE: A0 2C	1166	L0Y	WS2C	
FCCE: CA	1167	DEX		
FCCE: 60	1168	RTS		
	1169			
FCCE: A2 08	1170	R0YTE	L0X	WS08
FCCE: 48	1171	R0YTE2	PNA	
FCCE: 20 FA FC	1172	JSR	R0CE1T	

FCF2: 68	125	PLA		;Reconstitui acumulador do stack
FCF3: 2A	126	ROL		;Introduz pelo carry, o bit lido
FCF4: A0 3A	127	LDY #3A		;Y com 3A como contador de tempo
FCF6: CA	128	DEX		;Le proximo bit ate o 8
FCF7: 00 F5	129	ONE	ROBTE2	
FCF9: 60	130	RTS		
	131			
FCFA: 20 FD FC	132	ROBTE1	JSR ROBTE1	;2 meios ciclos de onda de TAPEIN
FCFB: 88	133	ROBTE1	DEY	;Decrementa Y ate transicao
FCFC: A0 1D C0	134		LDA #0TPIN	da onda de TAPEIN
FCFD: 45 2F	135		EDR LASTIN	
FCFE: 1D FB	136		BPL ROBTE1	
FCFF: 45 2F	137		EDR LASTIN	;Armaz. em LASTIN o ultimo estado
FD07: 05 2F	138		STA LASTIN	da forma de onda de TAPEIN
FD09: C0 80	139		CPY #80	;Se o bit=1, seta carry
FD0B: 60	140		RTS	
	141			
FD0C: 4C 38 00	142	IM0	JMP (RSL)	;Se teclado, indica SFDOF
FD0F: 4C 02 FB	143	IM1	JMP IM2	
FD12: 8A	144	CURE98	TAA	
FD13: FB 67	145		BEQ JIMS	;Se inicio do buffer, novos dados
FD15: 4C 98 F7	146		JMP PRERET	;Verif. caract. xiler. do buffer IN
FD18: CA	147	DECBUF	DEX	
FD19: 20 82 FB	148		JSR POSCUR	;Apaga o cursor
FD1C: 20 18 FC	149		JSR DESESA	;Desloca uma posicao a esquerda
FD1F: 20 82 FB	150		JSR POSCUR	;Reposiciona o cursor
FD22: 4C 7C FD	151		JMP JIMS	
FD25: 20 82 FB	152	SRCUR	JSR POSCUR	;Apaga cursor
FD28: 20 25 FB	153		JSR MXTPI81	;Desloca uma posicao a direita
FD2B: 20 36 FD	154		JSR PSOR	;Repos. o cursor e increm. buffer
FD2E: 00 4C	155		BNE JIMS	;Se nao completou buffer JIMS
FD30: 4C 30 FD	156		JMP POSCUR1	;Se completou buffer POSCUR1
	157			
FD33: FF FF	158	HEX	FFFF	
	159			
FD35: 4C DC FD	160	INTC	JMP DND	
	161			
FD38: 20 82 FB	162	PSOR	JSR POSCUR	
FD3B: E0	163	DND		;Incrementa posicao do Buffer
FD3C: 60	164		RTS	
	165			
FD3D: C9 88	166	POSCUR1	CMF #80	;Compara com (- ou CTRL-H
FD3F: FD 01	167		BEQ CURE98	;Se igual desloca a esquerda
FD41: C9 95	168		CMF #95	;Compara com (= ou CTRL-U
FD43: FD 0D	169		BEQ SRCUR	;Igual, desl. direita, incr. Buffer
FD45: C9 FD	170		CMF #FFD	;Compara com *
FD47: FD 0F	171		BEQ SRCUR	;Sobe cursor se igual
FD49: C9 F1	172		CMF #FF1	;Compara com
FD4B: FD 09	173		BEQ SRCUR	;Desce cursor se igual
FD4D: C9 98	174		CMF #98	;Compara com CTRL-X
FD4F: FD 13	175		BEQ JPBRES8	;Reinicia Buffer se igual
FD51: 4C 7E EF	176		JMP CARBUF	;Carrega no Buffer IN
FD54: 00 85	177	BSROUT	BNE BSROUT	;Nao complet. buffer, impr. caracter
FD56: FD 0C	178		BEQ JPBRES8	;Completo Buffer, reinicia Buffer
FD5B: 20 82 FB	179	SRCUR	JSR POSCUR	;Prepara cursor
FD5D: 20 ED FB	180	BSROUT	JSR CNUT	;Imprime caracter
FD5E: 20 82 FB	181		JSR POSCUR	;Reposiciona cursor

FD61:	4C 7C FD	182	JMP	JMS	:Testa entrada de dados p/teclado	
		183				
FD64:	20 20 FC	184	JMPRESA	JSR	PRINCSA	:Imprime "\n"
FD67:	20 8E FD	185	GETLAZ	JSR	CROUT	:faz um CR
FD6A:	A5 33	186	GETLN	LDA	PROMPT	:Impr. carac.de iniciat"0,1,"
FD6C:	20 ED FD	187		JSR	CONT	
FD6F:	AF A0	188	GETLH1	LDA	R" "	:Limpa buffer IH
FD71:	A2 90	189		LDX	#500	:X como cont. de compr. do buffer
FD73:	90 00 02	190	CLARIH	STA	IH,X	
FD74:	EB	191		IMH		
FD77:	00 FA	192		RNE	CLRIH	:Limpa buffer ate o fia
FD79:	20 82 FB	193		JSR	POSCUR	:Coloca o cursor no video
FD7C:	20 33 FC	194	JMS	JSR	DMS	:Verif.entrada de dados p/teclado
FD7F:	4C A2 EF	195	JMP	TSPRAL		:Se FQVL=0, continua em VFLIN
FD82:	C9 80	196	VFLIN	CMP	#560	:Compara com RETURN
FD84:	80 87	197		RNE	POSCUR1	:Se nao, faz nova comparacao
FD87:	90 00 02	198	ADDIMP	STA	IH,X	:Se e RETURN coloca no buffer IH
FD89:	EA	199		HOP		
FD8A:	EA	100		HOP		
FD8B:	20 84 FF	101	CROUT1	JSR	CLIREOL1	:Limpa linha a partir do cursor
FD8E:	AF 60	102	CROUT	LDA	#580	:Carrega codigo ASCII de RETURN
FD90:	00 58	103		RNE	CONT	:Saida de dados
FD92:	A4 30	104	PRAL	LDI	ALH	:Y=ASH
FD94:	A6 3C	105		LDX	ASL	:X=ASL
FD96:	20 8E FD	106	PRYX2	JSR	CROUT	:faz um RETURN
FD99:	20 40 FY	107		JSR	PRINTX	:Impri.Y e X hexadecimal
		108				: (Endereco contido em Ae)
FD9C:	A0 00	109		LDY	#500	
FD9E:	AF A0	110		LDA	R" "	:Imprime um traco "-"
FD9D:	4C ED FD	111	JMP	CONT		
FD93:	A5 3C	112	XAMB	LDA	AIL	:Ajusta A2 para imprimir
FD95:	09 07	113		ORA	#507	endereco de final 0 ou 8
FD97:	85 3E	114		STA	A2L	
FD99:	A5 30	115		LDA	AIM	
FD9B:	85 3F	116		STA	A2H	
FD9D:	A5 3C	117	MOORCH	LDA	AIL	:Verif.se e ender.de final 0 ou 8
FD9F:	29 07	118		AND	#507	
FD91:	00 03	119		RNE	DATAOUT	:Nao, imprime conteudo do endereco
FD93:	20 92 FD	120	XAM	JSR	PRAL	:Sim, impr.o ender. contido em AI
FD95:	AF A0	121	DATAOUT	LDA	R" "	:Imprime um espaco
FD98:	20 ED FD	122		JSR	CONT	
FD9B:	R1 3C	123		LDA	(AIL),Y	:Impri.conteudo do end.AI em hexa
FD9D:	20 8A FD	124		JSR	PRYTYE	
FD9D:	20 8A FC	125		JSR	MEXTAL	:Increment. AI, se AI=A2 seta carry
FD9D:	90 EB	126		BCC	MOORCH	:Verif. se e vez de impr. ender.
FD95:	60	127	RTSAC	RTS		
		128				
FD9A:	4A	129	XAMPH	LSR		:E modo XAM, ADD ou SUB
FD97:	90 EA	130		BCC	XAM	
FD99:	4A	131		LSR		:Verifica se e soma ou subtracao
FD9A:	4A	132	TSTPH	LSR		
FD9B:	A5 3E	133		LDA	A2L	:Carrega A com A2L
FD9D:	90 02	134		BCC	ADD	:Se for soma faz ADD
FD9F:	49 FF	135		EOH	#5FF	:Se sub, faz complemento 2
FD91:	A5 3C	136	ADD	ADC	AIL	:Soma AIL com carry ao acumulador
FD93:	4B	137		PHA		:Salva resultado no stack
FD94:	A9 60	138		LDA	R" "	:Imprime "=" seguindo o resultado

FD04: 20 ED FD	1139	JSR	COUT	
FD09: 48	1140	PLA		;Retorna resultado do stack
FD0A: 48	1141	PRINT	PRN	;Salvo byte hexa a ser impresso
FD0B: 4A	1142	LSR		;Transfere o DMS para o des
FD0C: 4A	1143	LSR		
FD0D: 4A	1144	LSR		
FD0E: 4A	1145	LSR		
FD0F: 20 E5 FD	1146	JSR	PRHEXZ	;Subrotina de impressao em Hexa
FD2: 48	1147	PLA		;Retorna byte
FD23: 29 0F	1148	AND	#50F	;Imprime o des em hexadecimal
FD25: 09 80	1149	PRHEXZ	ORA	MSB
FD27: C9 8A	1150	CHP	MSB	;Se for entre 0 e 9 impr.direto,
FD29: 90 02	1151	BCC	COUT	senao soma 6 e imprime letra
FD2B: 4F 04	1152	ADC	#606	de A a F
FD2D: 4C 3A 00	1153	COUT	JMP	(CSAL)
FD2F: 4C A6 F1	1154	COUT1	JMP	COUT2
	1155	PUT	PARTE25,02	
	11			
FD33: A0 CF C8	12	HEXTAB	ASC	" OK WAIT"
FD34: A0 07 C1 C9	DA			
	13			
FD3B: 99 EC	14	SAVEADR	DA	SAVE1-1
FD3D: A0 ED	15	LOADADR	DA	LOAD1-1
	16			
FD3F: FF	17	HEX	FF	
	18			
FE00: C6 34	19	RLI	DEC	YSW
FE02: F0 9F	110	BEQ	XAMB	;Volta uma posicao no Buffer IN,T
FE04: CA	111	BLANK	DEX	;Se pos.0, impr.continuo em Hexa
FE05: 00 16	112	RNE	SETNOZ	;Seta NOZE e termina BLI
FE07: C9 BA	113	CHP	#12	;Compara com "12"
FE09: 00 88	114	RNE	XAMFN	;Se nao, imprime endere e continuo
FE0B: 85 31	115	STOR	STA	NOZE
FE0D: A5 3E	116	LDA	A2L	;Sim, seta NOZE e transf. contendo
FE0F: 91 40	117	STA	(A2L),Y	de A2L para o endereco indicado
FE11: E6 40	118	INC	A2L	por A2
FE13: 00 02	119	RNE	RTSS	;Proximo endereco em A2
FE15: E6 41	120	INC	A3H	
FE17: 60	121	RTSS	RTS	
	122			
FE18: A4 34	123	SETNOZ	LDY	YSW
FE1A: 39 FF 01	124	LDA	IN-1,Y	;Salva cod.de (1), (*), (1-), (1)
FE1D: 85 31	125	SETNOZ	STA	NOZE
FE1F: 60	126	RTS		
	127			
FE20: A2 01	128	LT	LOX	#501
FE22: 85 3E	129	LT2	LDA	A2L,X
FE24: 95 A2	130	STA	AML,X	;X para transferir o DMS primeiro
FE26: 95 44	131	STA	ASL,X	;Copia 2 bytes de A2 em A4 e A5
FE28: CA	132	DEX		
FE29: 10 F7	133	BPL	LT2	
FE2B: 60	134	RTS		
	135			
FE2C: 81 3C	136	MOVE	LDA	(ALL),Y
FE2E: 91 A2	137		STA	(AML),Y
FE30: 20 84 FC	138	JSR	MTM4	;Incrementa A1 e A4 ate A1=A2
FE32: 90 F7	139	BCC	MOVE	;Se A1=A2 carrega proximo byte

FE351	40	140		RTS	
		141			
FE361	81 3C	142	WTF	LDA (A1L),Y	;Compara A1 com A4
FE381	01 42	143		CMF (A4L),Y	
FE3A1	FD 1C	144		BEB VFTDN	;Se igual compara proximo byte
FE3C1	20 92 FD	145		JSR PVAL	;Se !=, impr.ender. contido em A1
FE3F1	81 3C	146		LDA (ALL),Y	;Impr.byte contido em A1 em hex.
FE411	20 DA FD	147		JSR PRBYTE	
FE441	A9 AD	148		LDA #"	;Imprime um espaco
FE461	20 ED FD	149		JSR COUT	
FE491	A9 AD	150		LDA #(""	;Abre parenteses
FE4B1	20 ED FD	151		JSR COUT	
FE4E1	81 42	152		LDA (A4L),Y	;Imprime o byte contido no ender. A4 em hexadecimal
FE501	20 DA FD	153		JSR PRBYTE	;Fecha parenteses
FE531	A9 A9	154		LDA #")"	
FE551	20 ED FD	155		JSR COUT	
FE581	20 B4 FC	156	WTFOK	JSR KXTAA	;Incrementa A1 e A4 ate A1=A2
FE5B1	90 09	157		BCC WY	;Se A1=A2 entao verif.prox.byte
FE5D1	60	158		RTS	
		159			
FE5E1	20 75 FE	160	MLIST	JSR AIPC	;Move A1 p/PC, se end.especificado
FE611	A9 14	161		LDA #514	;Disassembla 20 instrucoes
FE631	4B	162	LIST2	PNA	
FE6A1	20 80 FB	163		JSR INSTOSP	;Imprime a instrucao
FE6F1	20 53 F9	164		JSR PCADJ	;ajusta PC para proxima instrucao
FE6A1	85 3A	165		STA PCL	
FE6C1	84 3B	166		STY PCH	
FE6E1	6B	167		PLA	
FE6F1	3B	168		SEC	
FE701	EY 01	169		SBC #501	
FE721	00 EF	170		RNE LIST2	;Disassembla ate a vigesima inst.
FE741	60	171		RTS	
		172			
FE751	8A	173	AIPC	TXA	
FE761	FD 07	174		BEQ AIPCRTS	;Se X=1, transfere A1 para PC,
FE7B1	85 3C	175	AIPCPL	LDA A1L,X	;Se X=0, nao transfere nada
FE7A1	95 3A	176		STA PCL,X	
FE7C1	CA	177		DEX	
FE7D1	10 F9	178		BPL AIPCPL	
FE7F1	60	179	AIPCRTS	RTS	
		180			
FE801	AD 7F	181	SETDOW	LDY #57F	;Carrega INWFLS com #57F,
FE821	00 02	182		RNE SETIFL6	produzindo video no modo inverso
FE841	AD 00	183	SETHORN	LDY #500	;Carrega INWFLS com #500,
FE861	84 32	184	SETIFL6	STY INWFL6	produzindo video no modo normal
FE8B1	AD	185		RTS	
		186			
FE8F1	A9 00	187	SETOR0	LDA #500	;Simula porta 0 em A2L para
FE8B1	85 3E	188	IMPORT	STA A2L	especificar teclado como entrada
FE8D1	A2 3B	189	IMPRT	LDX #CSWL	
FE8F1	AD 0F	190		LDY #2H1	
FE911	00 00	191		RNE IMPRT	
FE931	A9 00	192	SETVID	LDA #500	;Simula porta 0 em A2L para
FE951	85 3E	193	OUTPUT	STA A2L	especificar teclado como saida
FE971	A2 3A	194	OUTPUT	LDX #CSWL	
FE991	AD FD	195		LDY #CONT1	
FE9B1	A5 3E	196	IMPRT	LDA A2L	;Seta RAM VETORES IN/OUT

FE9D: 29 0F	>97	AND	#00F	;Testa des
FE9E: F0 06	>98	BEQ	IOFRT1	
FE9F: 09 98	>99	ORA	#98	;R99=(A)<(PF
FEA0: A0 00	>100	LDY	#00	;Y=000
FEA1: F0 02	>101	BEQ	IOFRT2	
FEA2: A9 F0	>102	IOFRT1	LDA	#F0 ;End.+ signif.das rotinas de I/O
FEA3: 94 00	>103	IOFRT2	STY	LOC0,X ;* Normalmente:
FEA4: 95 01	>104	STA	LOC1,X ;(CSWL)=#00F(INS)	
FEA5: 60	>105	RTS		; (CSWL)=#00F (COULT)
	>106			
FEA6: EA	>107	NSP		
FEA7: EA	>108	NSP		
	>109			
FE8D: 4C 00 C2	>110	JMP	BASIC	;Va para o BASIC inicializando
FE8E: 4C 03 C2	>111	BASCONT	JMP	BASIC2 ;Continua no BASIC
	>112			
FE84: 20 75 FE	>113	06	JSR	ASPC ;Carrega PC com conteúdo de AI
FE85: 20 3F FF	>114		JSR	IOFRTONE ;Restaura todos os registradores
FE86: 6C 3A 00	>115		JMP	(PCL) ;Executa no ender. indicado por PC
	>116			
FE8F: 4C 02 FA	>117	REGZ	JMP	REGOSP ;Impr.o cont.dos regs.A,X,Y,P,S
FE90: 60	>118	TRET	RTS	
FE91: EA	>119	NSP		
FE92: 60	>120	TRET	RTS	
	>121			
FE93: C1	>122	RTBL	ASC	"A"
FE94: D6	>123		ASC	"X"
FE95: 09	>124		ASC	"Y"
FE96: 00	>125		ASC	"P"
FE97: 03	>126		ASC	"S"
	>127			
FE8A: 4C FB 03	>128	USER	JMP	USRAOR
	>129			
FE84: A9 40	>130	WRITE	LDA	#540 ;Emite um sinal piloto de 10 seg.
FE85: 20 C9 FC	>131		JSR	HEADR
FE86: A0 27	>132		LDY	#527
FE87: A2 00	>133	WRL	LDX	#600
FE88: A1 3C	>134		EOB	(A1,X) ;Check Byte
FE89: 40	>135		PHA	
FE8A: A1 3C	>136		LDA	(A1,X)
FE8B: 20 ED FE	>137		JSR	WRBYTE ;Escreve um Byte de AI
FE8C: 20 BA FC	>138		JSR	NEXTAI ;Incrementa AI
FE8D: A0 10	>139		LDY	#510
FE8E: 68	>140		PLA	
FE8F: 90 EE	>141	BCC	WRL	
FE90: A0 22	>142		LDY	#522 ;Se AI(A2 pega novo Byte de AI
FE91: 20 ED FE	>143		JSR	WRBYTE ;Escreve um check sum
FE92: F0 40	>144		BEQ	BELL ;Termina com BELL
FE93: A2 10	>145	WRBYTE	LDX	#510 ;Escreve um byte
FE94: 0A	>146	WRBYT2	ASL	
FE95: 20 06 FC	>147		JSR	WRBIT ;Escreve 1 meio ciclo de 1 bit
FE96: 00 FA	>148		BNE	WRBYT2
FE97: 60	>149		RTS	
	>150		PUL	PANTEZT,02
	>151			
FE94: 20 00 FE	>152	CONOM	JSR	BL1 ;Interpreta o conteúdo do Buffer
FE95: 48	>153		PLA	

FEFA: 68	34		PLA	
FEFB: 00 44	35		BNE MONZ	;Retorna no inicio do modo monit.
FEFD: 20 FA FC	36	READ1	JSR R02BIT	;Busca sinal piloto de TAPEIN
FEFE: A9 16	37		LDA #16	;atraso de 3,5 segundos
FEFF: 20 CF FC	38	READ2	JSR READR	
FF00: 05 2E	39		STA CHISUM	;Inicializa CHISUM=05FF
FF01: 20 FA FC	110		JSR R02BIT	;Busca sinal piloto de TAPEIN
FF02: A0 24	111	NRD2	LDT #24	;Busca sinal de SYNC
FF03: 20 FD FC	112		JSR R02BIT	
FF04: 00 F9	113		HCS NRD2	;Loop ate encontrar
FF05: 20 FD FC	114		JSR R02BIT	;Segundo SYNC (meio ciclo)
FF06: A0 38	115		LDT #38	;Indexa Y para teste de 0/1
FF07: 20 EC FC	116	RD3	JSR R02YTE	;Le 1 Byte
FF08: 01 3C	117		STA (AHL,X)	;Coloca em Ai
FF09: 45 2E	118		EDR CHISUM	;Atualiza CHISUM
FF0A: 05 2E	119		STA CHISUM	
FF0B: 20 BA FC	120		JSR NEXTA1	;Incrementa Ai
FF0C: A0 35	121		LDT #35	;Compara Y para teste de 0/1
FF0D: 90 FD	122		BCC RD3	;Le outro Byte ate Ai=A2
FF0E: 20 EC FC	123		JSR R02YTE	;Le CHISUM
FF0F: C5 2E	124		CMF CHISUM	;Compara com CHISUM
FF10: F0 00	125		BEQ BELL	;Se igual gera um BELL e retorna
FF11: A9 C5	126	PRERR	LDA #C5	;Imprime a mensagem "ERR"
FF12: 20 ED FD	127		JSR COUNT	
FF13: A9 02	128		LDA #02	
FF14: 20 ED FD	129		JSR COUNT	
FF15: 20 ED FD	130		JSR COUNT	
FF16: A9 07	131	BELL	LDA #07	;Carrega codigo ASCII de BELL
FF17: 4C ED FD	132		JMP COUNT	;Produz o beep
FF18: A0 F3 07	133	INRESTORE	LDA STATUS	;Armazena STATUS no stack
FF19: 48	134		PMA	
FF1A: A5 45	135		LDA ACC	;Restaura ACC no acumulador
FF1B: 4C 3A FC	136		JMP RST01	
FF1C: EA	137		NOP	
FF1D: EA	138		NOP	
FF1E: 4C 90 FB	139	IOSAVE	JMP SAVE1	;Salva acumulador e regist. X
FF1F: 0C F2 07	140	SAVE3	STY YREG	;Salva Y em YREG
FF20: 08	141		PHP	
FF21: 68	142		PLA	
FF22: 00 F3 07	143		STA STATUS	;Salva regist.de status em STATUS
FF23: 0A	144		TSX	
FF24: BE F4 07	145		STX SPNT	;Salva stack pointer em SPNT
FF25: 08	146		CLO	
FF26: 60	147		RTS	
FF27: FF FF	148			
FF28: FF FF	149		HEX FFFF	
FF29: 08	150			
FF2A: 20 3A FF	151	MON	CLO	;Muda modo decimal
FF2B: A9 00	152		JSR BELL	;Gera um BELL
FF2C: 05 33	153	MONZ	LDA #33	;Ponto de inicio do modo MONITOR
FF2D: 20 67 FD	154		STA PROMPT	
FF2E: 4C 67 FF	155		JSR GETLHZ	;Obtem linha de entrada de dados
FF2F: 4C 67 FF	156		JMP ZM000	
FF30: 4C	157			
FF31: 20 E9 EE	158		HEX 4C	
FF32: 20 AA FF	159	ZM000	JSR ZM00	;Zera modo
FF33: 20 AA FF	160	NXTLIN	JSR GETLIN	;Busca numero hex. no Buffer IN

FF72: 04 34	161	STY	YSW	;Salva posicao de IH em YSW	
FF74: A0 17	162	LDY	#617	;Verif. se e uma das funcoes	
FF76: 06	163	CHRSCH	DET	;do monitor em CHRTL	
FF77: 30 E4	164	BRJ	NON		
FF79: 09 CA FF	165	CMP	CHRTL,Y		
FF7C: 00 F8	166	BNE	CHRSCH		
FF7E: 20 08 EE	167	JSH	TOSUM	;Encontrou funcao, entao armazena	
FF81: A4 34	168	LDY	YSW	o ender. da func. no stack	
FF83: 4C 4F FF	169	JMP	NXT1TH	;Reposiciona IH, busca novo dado	
FF86: 20 99 FC	170	CLREOL	JSH	CLREOL	;Completa uma linha com "espaco"
FF89: 20 50 FA	171	JSH	ZXY	;Interpreta buffer IH	
FF8C: 60	172	RTS			
	173				
	174			;Coloca um endereco de ate 4 digitos em A2	
	175				
FF8D: A2 03	176	DIG	LOX	#503	;des do acumulador no des de A2L
FF8F: 0A	177	ASL			
FF9D: 0A	178	ASL			
FF91: 0A	179	ASL			
FF92: 0A	180	ASL			
FF93: 0A	181	NXTBIT	ASL		;Passa o bit atraves do carry
FF94: 26 3E	182	ROL	A2L		;des de A2L para DNS de A2L
FF96: 26 3F	183	ROL	A2H		;DNS de A2L para des de A2H
FF98: CA	184	DEX			;des de A2H para DNS de A2H
FF99: 10 F8	185	RPL	NXTBIT		;Desloca ate completar 4 bits
FF9B: A5 31	186	NXTBAS	LOA	MODE	;Se nao modo zero, va para NXTBAS2
FF9D: 00 DA	187	BNE	NXTBAS2		
FF9F: 85 3F	188	LDM	A2H,X		;Se modo 0,carrega A1 e A0 com A2
FFA1: 95 3D	189	STA	A1H,X		
FFA3: 95 41	190	STA	A3H,X		
FFA5: E8	191	NXTBAS2	INX		
FFA6: F0 F3	192	BEQ	NXTBAS		
FFA8: D0 DA	193	BNE	NXTCHR		;Proximo caracter do Buffer IH
FFAA: A2 00	194	GETNUM	LOX	#500	;Zera A2
FFAC: 86 3E	195	STX	A2L		
FFAE: 86 3F	196	STX	A2H		
FFB0: 8F D0 D2	197	NXTCHR	LOA	IH,Y	;Busca caracter do Buffer IH
FFB3: C8	198	JMY			;Proxima posicao do Buffer IH
FFB4: 4F 8D	199	EOB	#500		;Transforma ASCII de 80-8F em 0-9
FFB6: CF 0A	1100	CMP	#50A		;Entre 0-9 processe o numero
FFB8: 90 D3	1101	BCC	DIG		
FFBA: 4F 86	1102	ADC	#500		;ASCII de C1-C6 transf. em FA-FF
FFBC: CF FA	1103	CMP	#5FA		;Entre FA e FF processe o numero
FFBE: D0 CD	1104	BCS	DIG		
FFC0: 60	1105	RTS			;Ret.se nao e um numero hex.
FFC1: EE F8 D0	1106	NXTCOO	INC	COORD	;Incrementa o par (COORD-H)
FFC4: D0 D3	1107	BNE	COORDS		
FFC6: EE FC D0	1108	INC	COORD		
FFC9: 60	1109	COORDS	RTS		
	1110				
FFCA: 8C	1111	CHRTL	HEX	8C	;F(CTRL-C)
FFCB: 82	1112		HEX	82	;F(CTRL-F)
FFCD: 8E	1113		HEX	8E	;F(CTRL-E)
FFCE: 82	1114		HEX	82	;F(CTRL-F)
FFCF: 8F	1115		HEX	8F	;F(V)
FFD0: C4	1116		HEX	C4	;F(CTRL-X)
FFD2: 82	1117		HEX	82	;F(CTRL-Y)

FFD1: A9	1118	HEX A9	;FCTRL-P)
FFD2: B8	1119	HEX B8	;FCTRL-B)
FFD3: A6	1120	HEX A6	;F(")
FFD4: A4	1121	HEX A4	;F(")
FFD5: 06	1122	HEX 06	;F("M)
FFD6: 95	1123	HEX 95	;F(")
FFD7: 07	1124	HEX 07	;F("M)
FFD8: 02	1125	HEX 02	;F("I)
FFD9: 05	1126	HEX 05	;F("L)
FFDA: F0	1127	HEX F0	;F("M)
FFDB: 00	1128	HEX 00	;F("B)
FFDC: E8	1129	HEX E8	;F("R)
FFDD: 93	1130	HEX 93	;F(")
FFDE: A7	1131	HEX A7	;F(")
FFDF: C4	1132	HEX C4	;F("CR)
FFE0: 99	1133	HEX 99	;F("SLAKE")
FFE1: B2	1134	HEX B2	;NASCANT-1
FFE2: C9	1135	HEX C9	;NUS-1
FFE3: BE	1136	HEX BE	;NUEZ-1
FFE4: C1	1137	HEX C1	;NRET-1
FFE5: 35	1138	HEX 35	;NVEY-1
FFE6: BC	1139	HEX BC	;NMCPT-1
FFE7: C3	1140	HEX C3	;NTRT-1
FFE8: 96	1141	HEX 96	;NOUTPT-1
FFE9: AF	1142	HEX AF	;NASTIC-1
FFEA: 17	1143	HEX 17	;NSETMO-1
FFEB: 17	1144	HEX 17	;NSETMO-1
FFEC: 28	1145	HEX 28	;NMOE-1
FFED: 1F	1146	HEX 1F	;NLT-1
FFEE: B3	1147	HEX B3	;NSETNMH-1
FFEF: 7F	1148	HEX 7F	;NSETTM-1
FFF0: 50	1149	HEX 50	;NMLIST-1
FFF1: CC	1150	HEX CC	;NWRITE-1
FFF2: B5	1151	HEX B5	;NBO-1
FFF3: FC	1152	HEX FC	;NREACH-1
FFF4: 17	1153	HEX 17	;NSETMOE-1
FFF5: 17	1154	HEX 17	;NSETMOE-1
FFF6: FS	1155	HEX FS	;NCRNH-1
FFF7: D3	1156	HEX D3	;NBRNH-1
FFF8: 5D FF	1157	DA MN	;VECTOR (NVECTOR ENTER)
FFF9: FB D3	1158	DA MH	;VECTOR (INTERRUPTED NAO NASCADA)
FFFA: 62 FA	1159	DA RESET	;VECTOR (RESET)
FFFB: 4D FA	1160	DA IRB	;VECTOR (INTERRUPT REQUEST)

--End assembly, 16128 bytes, Errors: 0

Symbol table - alphabetical order:

ASH	=530	ALL	=53C	ALPC	=5FE75	ASPLP	=5FE78
ALPRTS	=5FE7F	ASH	=53F	AZL	=53E	ASH	=541
A3L	=540	A4s	=546	A4H	=543	A4L	=542
ASL	=544	ABS	=50EC9	ACC	=545	ACMPA1	=5EE9E
ACMPA2	=5EEC7	ACMPA3	=5EEB9	ADD	=50A03	ADL	=50AED
AD2	=50AEB	ADD	=50B04	AD4	=50B0B	ADS	=50B14

ADD	=SF001	ADDACC	=SF0EF	ADDIMP	=SF086	ADDMM	=SC080
ADDEX2	=S0038	ADDEX3	=S0025	ADDEX4	=S0040	ADDEXP	=S0028
ADJEXP	=SF0B9	ADJIMB	=SEBFS	ADJMM	=S004F	ADJ	=SF0BA
AMPDR	=S00F5	AMD	=S027D	AMM?	=SEB87	AMED	=S07F0
AND	=S45	ANDSH	=S4A	AMITH	=S00ED	AMBAT	=S0438
ANYFOUND	=S048B	ANYLOP	=S0487	ANYMM?	=S0493	ARTPMT	=SFA
ANYSTR	=S0826	ANYTAR	=S48	ANYMA2	=S07E4	ARYMA2	=S07EB
ANYVAR	=S070C	ANYV00	=S07F3	ASC	=S09FF	ASCHEX	=S0C07
ASS	=S0192	ASSEMB	=S0189	ASSEMB1	=S0180	ASSEMB2	=S01C8
AT?	=SE5AC	ATROAC	=SF024	ATLOP	=SE27E	ATH	=S0280
ATNI	=SE3C0	ATK2	=SE3CE	ATM3	=SE3E1	ATHSER	=SE2EB
AXB	=SF0F5	ATTMT	=S0426	AtAFAC	=S0B4A	AtAFACS	=S0B4C
BADMM	=S0314	BAS2H	=S2B	BAS2L	=S2A	BASCOMT	=SFE83
BASH	=S29	BASIC	=S0280	BASIC2	=S0203	BASL	=S2B
BASLOC	=S47	BASLOC1	=S80	BELL	=SFF3A	BELL0	=SFB09
BELL1	=SF00D	BELL2	=SFB84	BRAF	=SF6F5	BILMM	=SE02E
BTLANCH	=SE029	BIGNB	=SE775	BKAND1	=SE770	BL1	=SF000
BLANK	=SFED4	BLENGTH	=S91	BLTU	=S0682	BLTU2	=S0689
BPL1	=S0B24	BPL2	=S0916	BPL3	=S0949	BPL4	=S0999
BREAK	=SFAAC	BREAKIN	=S044C	BKVL	=S0CF0	B8B	=S0363
BKROUT	=SF054	BTCMD	=S0C0E	BTCM1	=S0C0F	BUF	=S0460
BUSBIT	=S011D	BUSCOR	=SEF2B	BUSESP	=S0C34	BUSTOKEN	=S730
BYPASS	=S032C	CALL	=SE536	CARACT	=SF67F	CARBUF	=SEF7E
CART	=S02C9	CARTD	=S900D	CART3	=S9003	CART6	=S9006
CARTY	=S9009	CAT	=S08B1	CTB	=S770C	CEMMod	=SE024
CH	=S24	CHAR1	=SF994	CHAR2	=SF99A	CHARAC	=S0D
CHGRF	=S0C3A	CHICLS	=S01ED	CHICOM	=S01E6	CHICOM	=S04C4
CHOP	=S0F80	CHICFLT	=SEBFE	CHICFST	=SEB0D	CHICOM	=S0C5C
CHMM	=S0092	CHICPM	=S01E3	CHICSTR	=S0094	CHICSUM	=S0E
CHMT	=SEBEE	CHICTYP	=S0D0C	CHIVAL	=S0095	CHIGET	=S81
CHIGOT	=S87	CHISRCH	=SFF76	CHISTR	=S0F6D	CHITBL	=SFFCA
CHITXT	=SFDCD	CLEAR	=S09B1	CLEARC	=S09B3	CLEALZ	=SFC98
CLEOP1	=SFC46	CLEXEP	=SFC9D	CLBBTH	=SFD73	CLREOL	=SFC99
CLREGL1	=SFFB6	CLREOP	=SFC42	CLKEY	=SFD74	CLRSC2	=SFB38
CLRSC3	=SFB0C	CLRSCR	=SFB02	CLRTOP	=SFB36	CLVL	=SFI98
CHTARL	=S02D6	CHPMH	=SE07C	CHPDOME	=S02E9	CHPMH	=SE087
CHPLOP	=S02D2	CHPLSP	=SEF9B	CHCR	=SFI17	COH	=SFC
COOL	=SFA	COORIS	=SFFC9	COLCOUNT	=SFA	COLOST	=SE442
COLON?	=S0B62	COLOR	=S3D	COLOR	=SE2A3	COMBYTE	=S0A66
COMLED	=S0C5C	COMPAKE	=S010C	COMNAME	=SE663	COMDEF	=S0254
COM	=S0B66	COMINT	=S0A15	COMT	=S0B8B	COMTAGS	=S0195
COMTB	=SFAF3	COMTBCH	=S010B	COMTCR	=SFI0A	COMTTOM	=S0A50
COMPM	=S0CFD	COMTOM	=SFF5C	COMSTR	=S0C8F	COMY	=S0C0C
COS	=SE304	COSTBL	=SE93B	COUNTED	=SE1D1	COUNTM	=S1D
COUT	=SFD0B	COUTL	=SFD0D	COUT2	=SFI46	COUT3	=SE0C0
CPLTBIT	=SE0CC	CPLTNAME	=SE03A	CPMAKX	=S16	CPROP	=S0D0D
CPRTYP	=S89	CR	=SFC42	CRD0	=S0E2D	CRMA	=SFEFA
CROUT	=SFA8E	CROUT1	=SFD8B	CSHF72	=SE7FE	CRSHFT	=SE7FC
CSST1	=S0E0B	CSST2	=S0E0A	CSST3	=SE05A	CSST4	=SE073
CSST5	=S0D7C	CSST6	=SE08C	CSST7	=SE095	CSWL	=S36
CTRC?	=S0F91	CURESB	=SFD12	CURLM	=S75	CURLSV	=SFA
CV	=S25	CV2	=S0D9C	CVB	=SFBF3	CT	=S0779
DATA	=S0CBA	DATAFLG	=S13	DATAN	=S0CC8	DATABUT	=SFB66
DATIM	=S0F91	DATLM	=S7B	DATPTH	=S7D	DECHUF	=SFD18
DECOMOM	=SFA41	DECTBL	=SE18B	DECK1	=SFI0D	DEF	=S0A2D
DEL	=SE492	DELH	=SE3	DESC?	=S0D81	DESESR	=SFC18
DESESR1	=SFC2D	DESBNV	=SEFED	DEST	=S6D	DFTDTH	=S0S11

DIG	=5FF8D	DIN	=502F9	DIMRLG	=51D	DIMLUP	=50256D
DIMK	=50509	DTHK2	=5058A	DIR?	=5CEEC	DIRECT	=5CAFF
DISK20	=5F59D	DISPAG1	=5C05A	DISPAG2	=5C055	DIV	=5007B
DIYD	=5004F	DIY2	=500F8	DIY+230	=5C50G	DNTH	=50169
DOCHP	=5C020	DOERRNSD	=5C700	DOMATH	=5016R	DONTH	=50162
DOWE	=5C927	DORIENT	=5CERS	DOGPC	=5CE5A	DOWN	=5E005
DOWH1	=5E38B	DOWH2	=5E38D	DOWH3	=5E38A	DP1	=5EAC1
DP2	=5EAC7	DP3	=5EAE6	DPD0	=50FAL	DPFLG	=59B
DPL	=5E089	DPLEFT	=50FC3	DPLOC	=5E08B	DPRIGHT	=50FCC
DPK2000	=5F25F	DRAW	=5EAE9	DRAW1	=5E986	DRAG2	=5E9A7
DRAM3	=5E981	DRAM4	=5E98C	DRAMS	=5E9C9	DRAG4	=5E982
DRAWNT	=5EAA0	DSCLN	=5BF	DSCPTA	=5BC	DSCINP	=59D
DSK	=5C10D	DWRAC0	=5C252	DV1	=50052	DV2	=5035C
DVAR	=50B3D	DVARS	=50B33	DWARTS	=50B6C	DWH	=501
DRL	=50D	DY	=5A2	EDTMS	=5F6D4	EH	=5D5
EL	=5D4	END	=5C895	END2	=5C896	ENOS	=5C28D
END4	=5C8AF	ENDCHR	=5DE	ENDFOR	=5C07D	ENDWASH	=535
ENHIMP	=5D07A	ENHMO	=5C90B	ENDTBS	=5F707	EPDVID	=5F723
ENIC	=5E95A	EN3	=5E8AD	EN4	=5E88B	EBUL	=5D1F8
ENUMP	=501C0	ENFLG?	=5C88B	ENLIN	=5CEA4	ENLUP	=5C70E
ENR	=5F84B	ENRDR	=50A2D	ENRFLG	=5D8	ENRIN	=5C63F
ENRLIN	=5D6	ENRMN	=5DE	ENRME	=5C7D1	ENRPOS	=5DC
ERRSTK	=50F	ESPERA	=5F1E3	ESPRA6	=5A7FF	ESTON	=5C58F
ETOKEN	=5F753	EVAL	=50F7B	EVALIM	=5E5AA	EVD	=5D0D8
EVDOME	=50F03	EVAR	=5EAD1	EXCSTR0	=5C5FF	EXECUTE	=5C137
EXLG	=50067	EXTT	=50185	EXP	=5C72D	EXPIM	=58A
EXPGR	=5E1FA	EXPGRH	=59C	EXTD	=5F8BF	EXTD4	=5E733
EXTRAFAC	=5AC	EXTRASV	=592	EXCHEN	=5C596	FAC	=59D
FACSGR	=5A2	FACSTRMS	=5E05D	FAD0	=5DAB8	FAD0H	=5DAB4
FA001	=5DAD8	FALSE	=5D2B5	FC1	=5D703	FC2	=5D7D9
FCONP	=5DECC	FCONP2	=5DECE	FD1	=5D09A	FD2	=5D0B0
FD3	=5D08D	FD4	=5D0C0	FD5	=5D0CE	FD6	=5D0E0
FD7	=5D0EF	FD8	=5D0A2	FDIV	=5D0B0	FDIVT	=5D0B3
FE1	=5D72F	FE2	=5D739	FENH3	=5F706	FEH0	=5D711
FEVLOOP	=5D0AE	FEH1	=5F10A	FETHH4	=5E728	FENCAR	=5C2C1
FINDCC	=5F702	FINDSP	=5EFS7	FINFEND	=5C222	FINLIN	=5D482
FINDOTA	=5C263	FINDPA	=5F4FC	FINDMAE	=5E2D0	FINTENP	=5C294
FINTTRANS	=5F48D	FIN	=5DFA4	FIR2	=5DFA8	FIR3	=5DFF7
FINDATA	=5CFC0	FINDCOL	=5C8D0	FIRST	=5FD	FIRSEV	=5FA94
FL1	=5C935	FL2	=5C94C	FLBRN	=5D44D	FLD1	=5D0B5
FLD2	=5D0BA	FLD4T	=5D0AD	FLOP	=5D8AC	FRI	=5DCCA
FLR2	=5DCCF	FLR3	=5DCC2	FLN	=5DCEE	FRT1	=5F9A2
FRT2	=5F99A	FLN	=5DCA1	FLNLT	=5DCC9	FLNLT	=5DCC9C
FNP	=5D1CC	FNC?	=5D45B	FNCDATA	=5D4C9	FNCNAM	=58A
FNOBY	=5D483	FNOLEN	=5D5A5	FNOFOR	=5C659	FNOIN	=5C731
FNOVNR	=5D7A2	FON	=5CAB0	FON2	=5CAB1	FORNAT	=52E
FORINIST	=5C12D	FORINACT	=5C60C	FORINT	=5B5	FOUT	=5E04E
FPGEN	=5A4	FWRT	=5E1B1	FE1	=5D8BE	FR2	=5D89A
FR3	=5D8A7	FRE	=5D5F8	FR2	=5D5FF	FRFAC	=5D95A
FR4	=5C25A	FRESPC	=571	FRSTR	=5D917	FRTHP	=5D91E
FRTHG	=5D94F	FRTOP	=5AF	FRVCL	=5DCA3	FRVCL2	=5D0B0
FRVH4	=5D0BF	FRVND	=5D8A9	FSH	=5D4C1	FSURT	=5D4C4
FULL	=5D7BE	FUNCT	=5D66E	FUNDEF	=5C62C	GABAG	=5D79E
GAFLE	=513	GASCALC	=5F847	GASH	=527	GASL	=526
GBCALC	=5F85A	GDRUID	=5F7F8	GC	=5CA4B	GO	=5D413
GDRUFS	=5C94E	GERCHR	=5F2D0	GERBF	=5F4D0	GERB	=5D0C5
GET	=5C0C5	GETADR	=5D4AC	GETMY	=5D4D7	GETMY2	=5D4D9

GETARYPT=SE859	GETBYT =S0412	GETCHR =S0443	GETEXP =SE002
GETFNT =SF8AC	GETDN =SC0AC	GETLINK =SC955	GETLN =SF0A4
GETLMS =SF0AF	GETLN2 =SF0AF	GETNUM =SF0AA	GETOLD =S04C3
GETSPA =S07AC	GETSPC =S07AE	GETSTR =S09F6	GETVAL =S04B8
GEERR =SEAG7	GID =SC0A2	GIVAYF =S040C	GI =SC0B5
GGE =S0S86	GO =SFE86	GKI =SC07A	GK2 =SC07E
GICMO =SC042	GICMO2 =SC044	GICMO3 =SC052	GICMO =S0340
GIEFR =SESSA	GIEEX =S0160	GIOEX =S0F84	GIOG =S03E2
GIOG =S040C	GOLINE =SC0CA	GIMEMST =S007A	GIROUT =S0267
GOSTART =S03B8	GOSTROUT=S03	GOSUB =SC046	GITF08 =S0037
GOTO =SC0A3	GOTO? =SC019	GOTSPA =S0783	GOMNRN =S00
GR =SEAF9	GRI =SEAF1	GRAFFLG =S0452	GRRPAS =S037C
GRPLOT =SF001	GSE =S0S83	GSYMER =SC0A6	GSTRYC =S04BF
GTFORPNT=SCA54	GTLT =S032B	GTUIN =S0A60	H2 =S2C
HABLLIN =SF05A	HALF =SE17E	HAMOLDR=SE64A	HCLR =SE771
HCOLOR =SEASA	HCOLORS =S1C	HCOLRZ =SE4	HDSUHT =SE90C
HEADR =SFCC9	HFM0 =SE9AC	HFM01 =SE971	HFM02 =SE97F
HFMS =SEAC3A	HFMS1 =SE4AE	HGLIN =SE8B8	HGR =SE75B
HGR1 =SE74F	HGR2 =SE74B	HGRFLG1 =SF054	HERFLG2 =S0935
HISFFF =SEACF	HIDH08 =S94	HIGHTR =S96	HIDEN =SE3E7
HLIN =SE5B6	HLIN2 =SE801	HLIN3 =SE8E9	HLIN4 =SE91F
HLINE =SF819	HLINEL =SF81C	HLINUL =SE8B1	HMASK =S30
HMB1 =SES	HONE =SFC5B	HP2 =SEAB8	HP3 =SEABF
HP40 =SE6	HPLOT =SEAE7E	HPLOT3 =SE707	HPOSH =SE790
HPUSH1 =SE7C1	HPUSH2 =SE7C2	HTAB =SEB67	HTAB1 =SE84C
HTAB2 =SEB7A	HUM =SC020	IDCTP =SF1AF	IDFC08 =SF0F0
IDFECTL =SF0B1	IDEM =SF09E	ID =SC0CE	IDFORM =SC0AE
IDTBUE =SC0G6	IDTBIAT? =SCLFA	IMPBS =SF770	IMPCH =SF0E9
IMPHAM =SEEC1	IMPBSL =SCA22	IMP8 =SC020	IMPSP =SE045
IMPWAIT =SEET0	IN =S0200	IMO =SF00C	IMI =SF00F
IN2 =SF002	IN3 =SF014	IN4 =SF018	IMS =SFC33
INCHR =SC0A8	INCHMP =SCSF3	IND05 =SF6DA	INDERR =SCL88
INDERR1 =SC186	INDXL =SSE	INDOLEG =SCS88	INDPAS =SF709
INDX =S99	IND =S0A00	IND1000 =SE4E7	INDBAS =SE4E9
INDBAS3 =SE51A	INDCART =S9FF1	INDCPL =SC070	INDSUN =SE8C3
INIT1 =SF03B	INLIN =SC818	INLIN2 =SC810	INLINK =SC824
INLTN4 =SC82A	INLTNS =SC84B	INMSG =SCA47	INPDME =SC0EE
INPERR =SC0A8	INPF3M =SCFC1	INPORT =SF0E8	INPRT =SC033
INPFR =S7F	INPUT =SC0E7	INPUTOR=SC0E6	INPUTFLO=S15
INSL =S097A	IN52 =S0960	IN53 =S09B1	IN54 =S0962
IN55 =S098C	IN5G1 =SF885	IN5G2 =SF88F	IN5G3 =SF891
INSTR010=S0F08	INSTRLIN=S0709	INSTANT =SC0F3	INSTOSF =SF0B0
INSTONE =S0903	INT =S0F030	INTC =SF035	INTFL6 =S12
INTPMT =SE0A3	INTVAR? =S0340	INVAL10 =SC1B1	INVCUR =SF0C4
INVERSE =SE500	INVLG =S32	IOCOLOR =SC050	IOPEB =SC051
IOPT1 =SFE9B	IOPT1 =SFEA7	IOPT2 =SF0A9	IORESTORE=SFF3F
IOSEME =SFF4A	IOERR =S0A83	IO4 =SFAA0	IOALQCL =S03FE
ISCDAS =SF769	ISCHTC =S0C78	ISCT8 =SE070	ISCT8? =SE0A0
ISCTP =SEF96	ISDRF? =SEC2C	ISLETC =S0397	ISLET? =SE048
ISTOK? =SC0A7	ISD =SE015	JARY =S0360	JCAR16 =SF133
JCAR19 =SF1F3	JCOCR =SF82E	J010 =SE099	JGCR0F =SF7E7
JER =S0485	JER8 =S074C	JERROR =S00A0	JF =S02F2
JGCR0 =SC00E	JJMS =SF07C	JLNC =SC012	JN10 =SE092
JNCR =SF127	JN1 =SESF7	JNADR5 =S70	JN1 =S0050
JNCR0 =SF064	JNCR12 =SF78A	JNCR18 =SF1F9	JNCR =SF77A
JNCR000=SC231	JSY =SC044	JSYN =SE6BF	JSYN01 =SEBF2
KBOUT =SC000	KBTFIN =SC010	KBL =S2B	KWR0 =SE0EF

WORD1 =SEEGE	LASTIN =SZF	LASTOP =SR7	LASTPT =VS3
LOWET =SFRP3	LECH =SDE20	LECL =SEBDC	LEFT =SE7E7
LEFT1 =SE7F1	LEFT2 =SE7F6	LEFTSTR =S0974	LEN =S09F0
LENGTH =SZF	LET =S0D48	LET2 =S0D88	LEFTREAL =S03FC
LETSTR =SCD9F	LF =SFC66	LFTRT =SE7E5	LG2 =S0G65
LIN =S0CBF	LIMCOOR =S0350	LIMCOB =SEFF1	LIMDET =S0331
LIMSET =SC7E1	LIMNUM =S50	LIMPRT =SEDC3E	LIST =S039C
LIST2 =SFE63	LISTED =SCA38	LISTLOP =SCA15	LIMEM =S0C
LOAD =SCREE	LOADAOR =SDFD0	LOADT =SEDAE	LOCD =S00
LOC1 =S01	LOC2 =S02	LOC3 =S03	LOC4 =S04
LOCK =S06	LOG =SDCS8	LOGSER =SECC32	LOGSIS =SDCS6
LOGE =SE1F5	LOMEH =SE6D7	LOOP20 =SEF44	LOMTR =S9B
LR1 =SE7EE	LRUD1 =SEB33	LRUD2 =SEB34	LRUD3 =SEB44
LRUD4 =SEB4B	LRUDX1 =SEB1C	LRUDX2 =SEB1D	LSTLIN =SCA0E
LSTD =S0ADC	LT =SFE20	LT2 =SFE22	MA =SECD0
MAF =SDEB0	MAF2 =SEB82	MAHIMP =SECF10	MAHMLST =SC9C3
MAWARY =S04D8	MAWDIGIT =SE10B	MAWDAT =S041C	MAWSTR =SEB07
MAWKEND =SE174	MAWTRIL =SC2YC	MB =SEDD1	MEBERB =SCAFF
MEMSTZ =S73	MEMS =SFAF3	MEMSERED =SC54F	MENTAB =SDFD3
MFA =SDE6F	MFA2 =SDE73	MFL =S0433	MIZ =SEB35
MIDSTR =S094B	MIM =S0DF6	MIMPRT =SFEED	MIHUS =S0C3F
MISATCH =S0C9E	MIXT =S0422	MLIST =SFE5E	MLAAD =SEB92
MMDL =SFP9C	MMEHR =SFA00	MMHKL =SFB81	MMHKL2 =SFB85
MMHKL3 =SFBCC	MOBCHM =SFD4D	MOBE =S31	MOB =SE742
MOH =SFF5D	MOH2 =SFF61	MOHMM =S0321	MOHMOFF =S0C52
MOHOF =SEB7D	MOHRA =SEB80	MOHUF =SDE38	MOHZF =SDE38
MOHAF =SDE7D	MOHVM =SEB66	MOHVE =SFE2C	MOHESTR =S0390
MOHVE =SEBFD	MOHVE2 =SEB02	MOHVA =SDE6D	MOHVN =SDE13
MOHVS =S0BDE	MOHVT =SE6C8	MOHWF =SDE45	MOHVL =SDE20
MOHSTR =S0BFC	MP =SEBFB	MRD2 =SFFD4	MSAVE =SEB14
MSHBL =SEB33	MSLUP =S0C0A	MUL1 =S0C0D	MU2 =S050A
MU3 =S05F3	MUL10 =S0D53	MULT =S0C07	MURACK =SE133
MURYT =SCAB2	MUG6 =SED11	MURAM =S0796	MURRG =SC7C3
MUS2 =S09B4	MUG3 =S0P00	MUZP =SE46C	MURMID =S0C31
MUROR =S0317	MH =S094A	MUR =SDA29	MU1 =S0D49
MUR =S0116	MOP =SDFE2	MUR =SE6E9	NEEDDEX? =SE14D
NEB2 =S0B8E	MUGATE =SCE25	MUSFAC =S0B8B	MESMUR =S041B
NEBOP =SE1EA	MUR =SC96D	MURDESC =S0C6A	MELM? =SC7AA
NEHMOB =SE7FB	NEHSTT =SCAEC	MURVAR =S0386	MEXT =S0321
NEHTRAL =SFCBA	NEHTECH =SCE54	NEHTECOL =SFB5F	MHONC =SFD32
NID =SE42B	NH1 =SE68B	NH2 =SE602	NH3 =SE60B
NH4 =SE9D9	NH5 =SE6A5	NHREDAT =SC200	NH =S025D
NHNT =S04EF	NHNTA =S05AC	NHUSEF =SEFEF	NH1 =S028E
NHNT =S03FB	NHNSP =SEFFD	NHNTK =SFAA2	NH1 =S0361
NHLAN =SC2NF	NHNMAL =SE3D6	NHNSUCH =SC95E	NH7? =S018B
NHTRMO =S03A1	NHTRMOO =S0402	NHTRMATH =S015D	NHTRK =S0B5F
NHTRANCE =SESD4	NHTR2 =S01C7	NHTR3 =SEDA9	NHTRinstr =S0262
NHTRB =SEFF1	NHTRD1 =S07AD	NHTRIT =SFB95	NH =SE50F
NHTRSA =SFC2B	NHTRTL =SFB14	NHTRP =SFLC2	NH =S07F9
NHTR =S0B81	NHTRT =SECS8	NHTRXT =SE764	NHMLD =S0D6A
NHTRER =S0191	NHTRCMP =S0C0B	NHTRMTH =S0F	NHTRM =S07B8
NHTR =S0375	NHTR2 =S037F	NHTR =S03DC	NHTR =S038E
NHTRARY =S0442	NHTRASLOC =SE4FA	NHTRSAV =SECC7	NHTRYT =SC6A6
NHTRAMR =SCE69	NHTRAMR =SE3FF	NHTRM =S0C82	NHTRM =SCE33
NHTRCOO =SFFC1	NHTRCOL =SFD0E	NHTRD16 =S0C65	NHTRD16IT =S0F7E
NHTRDTH =S02F6	NHTRD =S059B	NHTROR =SC67A	NHTRM =S0CFD1
NHTRIMP =SCF16	NHTRKEY =SFB64	NHTRM =S074B	NHTRM =S0CFD

KOLHAI	=SEB80	KOLST	=SC9F1	KRM	=SE0A1	KRMUM	=SC025
KOPP	=S00FE	KOPAD	=SCA80	KRPG	=S05A8	KRPSOL	=SFBF0
KOPTR	=S0380	KXS	=SCFE1	KRPT	=S08F6	KRSPC	=SCES1
KOTA4	=SFCB4	KXTBAS	=SFF98	KXTBAS2	=SFFA5	KRTBIT	=SFF93
KOTRIT	=SFA8E	KXTCH	=SFFB0	KXTDTH	=S0446	KXTERR	=SE2B0
KXTITH	=S7FAF	KXTDPS	=SFB20	KXTPOS1	=SFB25	KXVAR	=S0373
KXY	=SFO30	KXZER00	=SEC8C	KXZER01	=SECCC	KXsef00	=SC5AF
KY	=SC889	KZ	=S0723	KxadDATA	=SC573	KXTEND	=SC130
KD00ER	=SE276	KDCKSUM	=SCE67	KL0LIN	=S77	KL0TEXT	=S79
KMCNT	=SC010	KMEDLY	=SFC22	KHERR	=SE62C	KNESTEP	=SCA00
KMG0T0	=SCD11	KOR	=SE23E	KR	=K0277	KRWASK	=S73
KOUT0	=SCEB1	KOUTFORM	=S0045	KOUTPRT	=SFE95	KTRPRT	=SFE97
KOUTUES	=SCE7F	KOUTROCH	=SFB21	KUTSP	=SCE7C	KVERFLOW	=S08EF
KPARCK	=S0104	KPARITY7	=K0DFF	KPARSE	=SCB81	KCADJ	=SFF53
KPCADJ2	=SFF54	KCADJ3	=SFF56	KCAJ4	=SFF5C	KCH	=S3B
KCL	=S3A	KCL	=SC271	KCLD	=SCD44	KCHH	=SCF74
KCEK	=S047E	KCHCHR	=SFC73	KCL	=S04B0	KCHCC	=SFB74
KCHWM	=SFCDA	KCL	=S0468	KCL2	=SCAAA	KCH	=S4B
KCAL	=S47	KHAOR	=SFE03	Kdiv2	=SE380	KLive22	=SC385
KLOT	=SES79	KLOT0	=SEFCA	KLOT1	=SEFC0	KLOT2	=SFB00
KLOTW8	=SES40	KLOTW	=SFB03	KLOT7	=SC918	KLOS	=SC3A0
KLUSEFS	=S08ED	KLUWAS	=SFA8E	KPTR	=S0F	KPSTK	=SE179
KLCC	=SFB7F	KPOINT	=S0A57	KPK	=S0A95	KPP	=SCC90
KOS	=S0619	KPSCH	=SFD0E	KPSCH	=SFB82	KPSCLR1	=SFB00
KPSE34	=SFB31	KPSP0	=SC080	KPREG	=SFAA5	KRAL	=SFB92
KPADR1	=SFB10	KPADR2	=SFB14	KPADR3	=SFB26	KPADR4	=SFB2A
KPADR5	=SFB30	KPDR34	=SFC20	KPBL2	=SFB4A	KPBL3	=SFB4C
KPRLAK	=SFB4B	KPRTTE	=SFBDA	KPROMO	=SFB86	KPROM1	=SFB08
KPREHC	=S011E	KPRETEST	=S00F5	KPRET	=SFB78	KPRER	=SFB20
KRESP	=SFB49	KPF1	=SCB80	KPF2	=SCB88	KPR1	=S0880
KPS12	=S0B12	KPS15	=S0B15	KPS18	=S0B18	KPS3	=S0B03
KPS6	=S0B06	KPS9	=S0B09	KPSG	=S0B0C	KPSGND	=S4F
KPSF	=S0B0F	KPMEX	=SFD03	KPMEX2	=SFD05	KPRINT	=SC3FA
KPRINT2	=SCDFC	KPLETR	=SFB78	KPMH1	=SFB85	KPMH2	=SFB89
KPRINTX	=SFB41	KPRINTL	=SFB08	KPRINTFAC	=SEDA8	KPRINTH7	=SC720
KPRINTOP	=SFB04	KPRINTX	=SFB44	KPRINTYX	=SFB40	KPROCESS	=SC15C
KPROCESS1	=SC17C	KPRCTCL	=SFB02	KPRG10	=SCC26	KPRHPT	=S30
KPROXCOM	=SC344	KPROXIN	=SFB61	KPROXOT	=SC258	KPROXOK	=SFB74
KPP	=SEB19	KPPREG	=SEC79	KPRTR	=SEDA8	KPRSTRNG	=SCDF4
KPTRK	=SCA6A	KPRTRSH	=SC059	KPRY2	=SFD96	KPSR	=SFD38
KPSH	=S0430	KPSHACK	=S0130	KPSHMD	=S0125	KPRDET	=SC003
KPTRSET2	=S0308	KPTRSET3	=SC00A	KPULL3	=SCCEA	KPUSHFAC	=S0140
KPUTCH	=SDF67	KPUTP	=SEB0A	KPUTHP	=S074F	KPUTX	=SE150
KPUTH	=SCB84	KPUTLNB	=SC7FE	KPUTHEW	=S0744	KPUTSTR	=SCD40
KPUTOK	=SCB82	KPW1	=SE18A	KW2	=SE104	KPWEDOP	=SC3F4
KPWUP	=SFAA8	KRWNT	=S03	KTL	=SFB20	KTL	=SFB2C
KQNT	=S0F0C	KROUT	=SCDE9	KQUARTER	=SE38A	KQ07	=S071E
K1AK	=S07FF	K3H	=S07	K3L	=S06	K4BK	=S07FF
K4H	=S09	K4L	=S0B	K4R	=S04F8	K4RY	=S05B2
KD1	=SE2E6	KD2	=SE301	KD2BIT	=SFDFA	KD3	=SFF16
KDIT	=SFCF0	KDNYTE	=SFDCC	KDNYTE2	=SFDCE	KDSP1	=SFA09
KDAD	=SFCB7	KDAD2	=SFF02	KDADNR	=SCDAD	KDADH	=SFF0D
KDADON	=SCA02	KDADL	=SE72C	KDADP	=SFFC0	KDADG	=S0017
KDADSP	=SFACC	KDADZ	=SFF0F	KDADN	=SFF38	KDADL	=SC100
KDADL1	=SC10C	KDADL2	=SC116	KDADN	=SC119	KDADN	=SCD01
KDADY	=SCBFF	KDADN	=SCC08	KDADN	=SFB	KDADN	=SCD00
KDADY	=SFA62	KDADN1	=SFA6E	KDADN	=SCD0C	KDADY	=SC728

RESTORE	=SCM49	RESALT	=S42	RESUME	=SE479	RET1	=SC481
RET10	=SC227	RET11	=SD006	RET12	=SD098	RET2	=SC4FE
RET3	=SC95F	RET4	=SC940	RETS	=SC877	RET6	=SC804
RET7	=SCC8F	RET8	=SCCC7	RETS9	=SCD30	RETCMA	=SF7C8
RETN1	=SCD50	RETG2	=SE0ME	RETH0	=SEEB7	RETHM	=SF7B1
RETH5	=SF941	RETST	=SCY88	RETURN	=SCCA9	RETHMR	=SC5C5
RODSPI	=SFACF	RIGHT	=SEB0A	RIGHISTR	=SD940	RN1	=SCD03
RK2	=SCC06	RNMEN	=S20	RND	=SE2C8	RNDALJ1	=SE2C0
RNDALJ2	=SE2CA	RND8	=SEBEC	RNDH	=S4F	RNDL	=S4E
RNDSEED	=SC9	ROT	=SEAA1	ROT2	=SF9	ROUM	=SE340
ROUND	=SEB94	RPSCH1	=SF8BA	RP264	=SF110	RS1	=SCA0C
RS2	=SCAED	RS3	=SCAEB	RSTR1	=SFC3A	RT1	=SF07E
RT2	=SF07F	RTNL	=SFECS	RTRC	=SF7FA	RTCS	=SF7F7
RTMP1	=SF0	RTMSK2	=SFA68	RTN1	=SD3A0	RTN10	=SC549
RTN11	=SF2F8	RTN12	=SDF63	RTN13	=SE1FA	RTN14	=SE2BF
RTN15	=SE3E7	RTN16	=SC40C	RTN2	=SDSC6	RTN3	=SD95F
RTN4	=SD489	RTN5	=SDR87	RTN6	=SDNEE	RTN7	=SDCFE
RTN8	=SD669	RTN9	=SDEBR	RTNwGSR	=SC563	RTS1	=SE809
RTS2	=SE981	RTS28	=SFBEF	RTS3	=SEAT0	RTSA	=SFC2C
RTS48	=SFC0B	RTS4C	=SFCDS	RTS5	=SFE17	RUN	=SCC37
RUMLINE	=SCC40	SAMEFOR	=SC66E	SAW2	=SF89F	SAV3	=SFF40
SAW01N	=SD501	SAW6	=SCB05	SAWE1	=SF890	SAWADR	=SFD0F
SAVET	=SE09A	SAWULB	=SD492	SAW8F	=SD0FF	SAVY	=SE12D
SAWYX	=SF070	SCALE	=SEAA7	SCALEZ	=SE7	SCAH1	=SF043
SC0CH	=SD357	SC0ND	=SF0E1	SCREEN	=SD221	SCRL1	=SF140
SC0RL2	=SF16B	SC0RL3	=SF171	SCRL4	=SF173	SCRM	=SF889
SC0R2	=SFA65	SC0RLL	=SF13A	SCRTCH	=SC962	SE	=SC880
SELEFAX	=SF7EA	SEND	=SCEB9	SEMDCH	=SCA19	SENE6	=SC77R
SEPTAP	=SF905	SERIES	=SE2BC	SERLEN	=S43	SELOOP	=SE2A3
SERMATN	=SE290	SERPNT	=S40	SE14	=SC2C2	SE1442	=SE392
SET4R	=SE4C1	SETCFLG	=SC2R6	SETCOL	=SF864	SETCR	=SF1C7
SETCTL	=SF9D0	SETDA	=SCB73	SETDP	=SDFB2	SETEND	=SC6A6
SETF04	=SE4E1	SETHJ	=SE5FA	SETHP0	=SE769	SETFLG	=SFE06
SETIMV	=SFE80	SETKB0	=SFE89	SETHQ2	=SFE1D	SETHW0	=SFE1B
SETHW0N	=SFEBA	SETPLP	=SFAA0	SETPTRS	=SC97C	SETSGN	=SDP9F
SETV10	=SFE93	SETVPNT	=SD3FB	SETWNO	=SFR4C	SFS	=SDC0D
SFSG	=SE058	SGN	=SDEAA	SEN7	=SD103	SEMOCK	=SDF46
SGNCPR	=S4B	SGWFLT	=SD61B	SENIF	=SD043	SHL	=SDC17
SH2	=SDC10	SH4	=SDC28	SHAFEN	=S1B	SHAPDL	=S1A
SHAFEPNT	=SE8	SHFTIN	=SC900	SHFTR	=SCC21	SHFTRES	=SD0F4
SHFT1	=SDCDA	SHIN	=SC909	SHL040	=SEAF5	SH	=SE3E1
SH1	=SE330	SH2	=SE340	SH3	=SE340	SHN	=SDFCF
SH0N1	=SEAC0	SH0N2	=SDCA2	SHENFLG	=S16	SHINF	=SD04B
SHM	=SE308	SHMGR	=SE3BF	SHINTER0	=SC558	SH2	=SC914
SKIP	=SD18C	SKPTOK	=SC910	SKP	=SCF77	SKPTK	=SCA6B
SKPV	=SD02A	SL1	=SEB16	SL2	=SEB2C	SL3	=SEB23
SL1	=SE71F	SL08P	=SFABC	SNDCMR	=SCA1C	SNDR7	=SDF47
SNTXERR	=SD135	SOFTEN	=SD0F3	SOFTVAL	=SD0F2	SNW0	=SC240
SPC7	=SDCA6	SPEED	=SESC9	SPEEDZ	=SF1	SPKR	=SCD30
SPNT	=SD074	SRR	=SE1A7	SRR4015	=SDCAC	SRRM11	=SCC47
SPCJR	=SF058	SROUT	=SF058	SRZCUB	=SFD05	SS	=SE29F
SPF	=SCBF7	ST1	=SD1R2	STACK	=SD10D	STATUS	=SCD73
STC0N	=SE519	STE	=SE07A	STEP	=SCAC9	STEX	=SEDIJ
STCMEN	=SE401	STITLE	=SFB68	STKINI	=SC994	STMP	=SE0B3
STR0N6	=SF6FC	STOP	=SCB93	STRN	=SFE0B	STONE	=SE70F
STR	=SD60F	STRCMP	=SD245	STRENO	=S40	STRINC	=SD4EF
STRIF	=SCB56	STRLLT	=SD7D1	STRLT2	=SD7D7	STRNGL	=S4B

STRING2	=S40	STRING7	=S0336	STROUT	=SC25F	STRPNT	=SC262
STRSPA	=S0AF7	STRTHG	=SC9CE	STRTXT	=S01A9	STXP	=SC2AF
STXTPT	=SC9AE	SUBERR	=S0480	SUMFLG	=S14	SUNMAN	=S0E20
SUMTBL	=SFFE1	SUMR	=S070F	SUMR0	=S0707	SUMS	=S07C3
SVID	=SEF82	SVR	=S0492	SVY	=SE005	SVYS	=SFD20
SWPNT	=SC7A6	SYNCH	=S04E8	SYNCH	=S01F1	TARGET	=SFE04
T40K	=SE4A9	TAB	=SC22B	TABAC	=S7590	TABCOR	=SFEAA
TABCSCH	=SFA70	TABIT	=SC250	TABV	=S7B5C	TABHERE	=SC23B
TABIMST	=SC118	TAN	=SE354	TAM2	=SE37C	TAFOUT	=SC020
TAPEPNT	=SEB3C	TENCART	=SC759	TEMP1	=S93	TEMP2	=S98
TEMP3	=S8A	TEMPPT	=S52	TEMPST	=S55	TEMO	=SE1A7
TEPBL	=SEFB4	TEPFLG	=SECB4	TEST41	=S0441	TEST42	=S0442
TEST50	=S0450	TEST51	=S0451	TEST53	=S0453	TEST70	=S0470
TEST71	=S0471	TEST72	=S0472	TEST73	=S0473	TEST74	=S0474
TEST84	=S0486	TEST86	=S0480	TESTCAR	=SC2A8	TESTCL	=SFD30
TEXT	=SE703	TEXT0	=SFDCE	TFN	=SEDF9	TIFED0	=SC10E
TIPED01	=SC1F8	TIPED02	=SC1F9	TITL0	=SFD97	TK2000	=S0CF9
TRPXP	=S99	TOK?	=SC29F	TOKNAME	=SCA79	TKEM?	=SCA4R
TOKLP	=SCA63	TOKLUP	=SCA6F	TOKTABL	=SC3B0	TONAL	=SC2AF
TOSUM	=SEED0	TOSUB1	=SEEB8	TRACE	=SE504	TRACE?	=SC1AF
TRANSBUF	=SFA80	TRDPLG	=SFD	TRET	=SFE2C	TRUE	=S020R
TRUE?	=SCDF0	TSA45	=SFD2A	T8B	=SEEDC	TSCART	=S9FF0
TSPBL	=SEFA2	TSTRP	=SFDCA	TSTL	=S0435	TUAB	=S07BA
TW08X	=S03B4	TXPSV	=S87	TXTPSV	=SFA	TXTPTR	=S8B
TXTTAB	=S47	ULTCHR	=SFA00	ULTVF	=SFD7F	UNARY	=S0234
UNDERR	=SCCA1	UNFNC	=S062B	UNFNC	=SC356	UNOT	=SC3BA
UP	=SFC23	UP0	=SEB55	UP1	=SEB68	UP2	=SEB7R
UP3	=SEB7D	UP4	=SEB7F	UPS	=SEB76	UPBLDC	=SFE12
UPDOWN	=SEB53	UPDOWN1	=SEB61	USER	=SFECA	USR	=S0A
USRADN	=S0CF8	V2	=S20	VAL	=S0A21	VALIDED	=SC5B0
VALTYP	=S11	VAR?	=S0194	VARL	=S01FD	VARNAM	=S81
VARYCT	=S0027	VARYNT	=S93	VARTAB	=S47	VARTID	=SCC15
VCTR	=S49	V00ME	=S0B77	VFCOM	=SC13B	VFLDN	=SFD82
VFY	=SFE36	VFYOK	=SFE2B	VIDOUT	=SFC05	VIDWAIT	=SFB70
VL2	=S0A29	VL3	=S0A41	VLIN	=SEB95	VLINE	=SFB2B
VLMEZ	=SFB26	VOLTRUF	=SFD7F	VPNT	=S40	VR1	=S0200
VR2	=S021E	VRFP	=SECA8	VSEARCH	=S03A9	VTAB	=SE580
VTRFE	=SFD70	VTRST	=SFB99	WAIT	=SFCAB	WAIT2	=SFC49
WAIT3	=SFC1A	WAITB	=S0A9E	WHAIX	=SE13C	WOBTH	=S23
WHLFT	=S20	WHDOT	=S22	WNEUP	=SE171	WNOBTH	=S21
WNR	=SCF9A	WGRN	=SEBDA	WR1	=SFE04	WREIT	=SFC06
WRIT2	=SFE07	WRITTE	=SEED0	WRITE	=SFC00	WRITPE	=SFC25
WIZ	=S04A0	WIZ	=S0A81	X0H	=SE1	X0L	=SE0
X1	=SE233	X2	=SE241	X3	=SE251	XAN	=SFD03
XANB	=SFD43	XAPFN	=SFD06	XBASIC	=SEED0	XBRK	=SFA59
XDRAN	=SEAEF	XDRAND	=SEB0E	XDRAN1	=SE9E2	XDRAN2	=SFA03
XDRAN3	=SEAD0	XDRAN4	=SEAA4	XDRAN5	=SEAZ5	XRES	=S07F1
XSHV	=S040E	YD	=SE2	YCNIC	=SEF	YRES	=S07F2
YSAV	=S34	YSAV1	=S35	ZARY	=S0234	ZEPNAME	=SEF1C
ZEND	=S0722	ZERDLY	=SFD0B	ZERO	=S004B	ZEROFAC	=S0B60
ZF	=SCF0E	ZFAC	=S0F5A	ZORAFLO	=SFA67	ZLUP	=S0543
ZNOO	=SEDE9	ZNOOG	=SFFAC	ZNOO1	=SEDEE7	ZPG4	=SFD1C
ZPGN	=SC208	ZPSTWFF	=SE425	ZTUTL	=SEF77	ZXY	=SFA50
at	=SC5	bell	=S87	data	=S83	equal	=S00
fn	=SC2	for	=S81	gsub	=S80	goto	=S4B
graf	=SFD	leftstr	=SEB	minus	=SC9	not	=SC6
plus	=SCB	pop	=S61	print	=S8A	ren	=S82

scrn	=507	sgn	=502	spc	=5C3	stop	=5C7
tab	=5C0	then	=5C4	to	=5C1		

Sybol table - numerical order:

DOWNH	=500	LOC0	=500	LOC1	=501	LOC2	=502
LOC3	=503	GOSTROUT	=503	LOC4	=504	R3L	=506
R3H	=507	R4L	=508	R4H	=509	USR	=50A
CHARAC	=520	ENDCAR	=50E	MUMDIA	=50F	PNTR	=50F
DIRFLG	=510	VALTYP	=511	INTFLG	=512	DATAFLG	=513
GRFFLG	=513	SUBFLG	=514	DAPUTFLG	=515	CPRMASK	=516
STRMFLG	=516	SHAPFL	=51A	SHAPEN	=51B	MODLOR1	=51C
COUNTH	=510	WMDLFT	=520	WMDWTH	=521	WMDTYP	=522
WMDSTH	=523	CH	=524	CV	=525	GBASL	=526
GBASH	=527	SASL	=528	BASH	=529	BASZL	=52A
BASZSH	=528	LNEN	=52C	H2	=52C	V2	=52D
RMEN	=520	FORMAT	=52E	CHOSUM	=52E	LASTEM	=52F
LENGTH	=52F	HWKX	=530	COLN	=530	MODE	=531
INTVFLG	=532	PRDPT	=533	ISAM	=534	EXOMASK	=535
YSWVL	=535	CSAL	=536	ISAL	=53B	PCL	=53A
PCH	=538	AIL	=53C	ASH	=53D	AZL	=53E
A2H	=53F	A3L	=540	ASH	=541	AAL	=542
A4H	=543	ASL	=544	ACC	=545	A46	=546
PVAL	=547	PEVN	=548	UCTR	=549	ENOL	=54E
RHSH	=54F	LTRMUM	=550	TEMPPT	=552	LASTPT	=553
TEMPST	=555	INDEX	=55C	DEST	=560	RESULT	=562
TXTTAB	=567	BASLOC	=567	VARTAB	=569	ARITAB	=56E
STREND	=56D	FRETOP	=56F	FRESFC	=571	HEMSIZ	=573
CURLIN	=575	BLDLEN	=577	DLTEXT	=579	DATLEN	=57B
DATPTL	=57D	INPTR	=57F	VARNAM	=581	for	=581
WARPMT	=583	data	=583	FORPMT	=585	TXPSV	=587
LASTOP	=587	bell	=587	CPRTYP	=589	FROMAM	=58A
TEMP3	=58A	OSCPTR	=58C	OSCLEN	=58F	JMPADRS	=590
RELEWTH	=591	EXTRABV	=592	TEMP1	=593	KHGRDS	=594
ARYPMT	=594	HGHTN	=596	TEMP2	=598	DMX	=599
TEMP2	=599	EXPON	=59A	DPFLG	=59B	LOWTR	=59B
EXPSON	=59C	FAC	=59D	DSCTMP	=59D	VPWT	=5A0
pop	=5A1	FACSGN	=5A2	SENLEN	=5A3	FRSEN	=5A4
ARG	=5A5	ARGSGN	=5A6	SOMCPR	=5A8	STRNG1	=5A8
goto	=5A8	EXTRAFAC	=5AC	SERPMI	=5A0	STRNG2	=5A0
FRSEMD	=5A7	BASLOC1	=5B8	gosub	=5B0	CHGET	=5B1
rea	=5B2	CHSGOT	=5B7	TXTPTR	=5B8	print	=5B8
tab	=5C0	to	=5C1	fn	=5C2	src	=5C3
then	=5C4	at	=5C5	not	=5C6	step	=5C7
plus	=5C8	ENDSEED	=5C9	alias	=5C9	DEL	=5C0
equal	=5D0	ODH	=5D1	DI	=5D2	sgn	=5D2
ORINT	=5D3	EL	=5D4	EN	=5D5	LOCK	=5D6
scrn	=5D7	DIRFLG	=5D8	DIRLIN	=5DA	ERRPOS	=5DC
ERRNUM	=5DE	ERRSTK	=5DF	XOL	=5E0	XOH	=5E1
IO	=5E2	DELH	=5E3	MODLOR2	=5E4	MODX	=5E5
IPAB	=5E6	SCALEZ	=5E7	SHAPENIT	=5E8	leftstr	=5E8
CONLCOUNT	=5EA	YDTC	=5EF	FIRST	=5F0	SPEEDZ	=5F1
TRDPLG	=5F2	graf	=5F2	GNMASK	=5F3	TXTPSV	=5F4
CONLSV	=5F6	KEYSTL	=5F8	ROTZ	=5F9	COOL	=5FB
COON	=5FE	RTDPL	=5FD	STACK	=50100	IN	=50200
BRNVL	=500FD	SOFTENL	=503F2	SOFTENH	=503F3	PUREDP	=503F4

AMPER	+503F5	TK2000	+503F8	USKADR	+503F8	WMT	+503F8
UNLOCAL	+503FE	P6A	+50400	TSITLL	+50435	FLGRAM	+50440
TEST41	+50441	TEST42	+50442	TEST50	+50450	TEST51	+50451
GRAFFLE	+50452	TEST53	+50453	RUF	+50460	PGR1	+50468
PGR2	+5046A	TEST70	+50470	TEST71	+50471	TEST72	+50472
TEST73	+50473	TEST74	+50474	FINDLN	+50482	TEST86	+50486
TEST80	+50480	XSAY	+5048E	ING	+50600	ARE6	+507F0
AREG	+507F1	YREG	+507F2	STATUS	+507F3	SPWT	+507F4
R16X	+507FF	PR61	+50800	PR03	+50803	PR06	+50806
PR09	+50809	PR6C	+5080C	PR0F	+5080F	PR12	+50812
PR615	+50815	PR618	+50818	ESPR16X	+547FF	R48X	+507FF
CART0	+59000	CART3	+59003	CART6	+59006	CART9	+59009
HERFLG1	+59854	HERFLG2	+59855	TSCART	+59FF0	INLCART	+59FF1
KR0UT	+5C000	KRTPIN	+5C010	TAPEOUT	+5C020	SPNR	+5C030
BOCDOR	+5C050	LWFO	+5C051	NOTADFF	+5C052	DISPA61	+5C054
DISPA62	+5C055	PRTSBRH	+5C059	BTCHT0	+5C05E	BTCHT1	+5C05F
POLD	+5C064	UNCPOL	+5C070	PRF1	+5C080	PRF2	+5C080
DSK	+5C100	RELAT	+5C100	RELAT1	+5C10C	RELAT2	+5C116
RELERR	+5C119	TAMINGST	+5C11B	BUSBYT	+5C11D	FORMINST	+5C120
VFCOM	+5C138	DETEAD	+5C150	PRONCON	+5C144	PROCESS	+5C15C
PROCESS1	+5C17C	INDERR1	+5C186	INDERR	+5C188	ASS	+5C192
CONFASS	+5C195	IMWALD	+5C181	ASSEMB	+5C189	ASSEMB1	+5C180
ASSEMB2	+5C1C0	CONTRCH	+5C1D8	TIFEND	+5C1DE	TIFEND1	+5C1F0
TIFEND2	+5C1F9	IMEDIAT7	+5C1FA	LMFR	+5C200	ZPDH	+5C208
MIMEDIAT	+5C200	FINFEND	+5C222	JPROCESS	+5C231	BUSESP	+5C234
SOUND	+5C240	TOMAL	+5C24F	OURACAO	+5C252	FREB	+5C25A
FDMOFA	+5C263	PROXNOT	+5C268	PBL	+5C271	SEMED	+5C278
FINTOMP	+5C284	TESTCAR	+5C2A8	MOLAN	+5C2AF	SETCFLG	+5C28E
FINDAM	+5C2C1	SET4	+5C2C2	GART	+5C2C9	RASIC	+5C203
BASIC2	+5C283	CMDTAML	+5C2D6	UNFMC	+5C256	MATHTRL	+5C392
KIDUS	+5C3A7	UNOT	+5C39A	PLUS	+5C340	TOTAML	+5C380
MENSERNO	+5C54F	MKEFOR	+5C54F	STATERNO	+5C55B	RTHSGSB	+5C563
MNDATA	+5C573	UNLDEE	+5C583	ESTOR	+5C58F	ExcMEN	+5C596
COMDEF	+5C5A4	INDILES	+5C588	REDIMMR	+5C5C5	DTVre200	+5C585
CONTILE6	+5C5E5	IMCOMP	+5C5F3	EXESTRG	+5C5FF	FORMuCX	+5C60C
IMPOSSL	+5C622	FINMDEF	+5C62C	ERRIN	+5C63F	IMH56	+5C647
BREAKIN	+5C64C	STFORPNT	+5C654	FMDFOR	+5C659	SAHEFOR?	+5C66E
HYFOR	+5C67A	RET1	+5C681	RLTU	+5C682	RLTU2	+5C689
SETEND	+5C6A6	MRYT	+5C682	MORYT	+5C686	KOPAG	+5C680
CHRMEN	+5C6C5	REASON	+5C692	RSI	+5C6DC	R52	+5C6ED
RS3	+5C6E8	RET2	+5C6AF	MORRR	+5C6FF	ERROR	+5C701
OCENRMS	+5C708	ERLUP	+5C70E	PRINTP?	+5C720	RESTART	+5C728
NOLIN	+5C748	ML1	+5C73E	MKNW	+5C736	NEULN?	+5C7A4
HYPRG	+5C7C0	INSTRLIN	+5C7D9	LINUSEY	+5C7E1	NOLINK	+5C7ED
PULLINK	+5C7FE	FINDOOL	+5C800	INLIN	+5C818	INLDO2	+5C810
INLINK3	+5C824	INLINA	+5C82A	INLINK5	+5C848	GOUFS	+5C84E
STRIP	+5C854	MO1	+5C861	INCR	+5C868	GETIN	+5C84E
PARSE	+5C861	KOCHR	+5C882	SE	+5C880	TIM?	+5C89F
ISTON?	+5C8A7	MY	+5C889	KX	+5C88E	LIN	+5C88F
PUTTOK	+5C8E2	PUTIN	+5C8E4	SSF	+5C8F7	REMY	+5C8F9
SHFTIN	+5C900	SHIN	+5C909	SKEPTOK	+5C910	SE2	+5C914
PUPP	+5C918	DOHE	+5C927	FINDLIN	+5C931	FL3	+5C935
FL2	+5C94C	DETLINK	+5C953	MOGLUCH	+5C95E	RET3	+5C95F
NEW	+5C968	SCRTOH	+5C962	SETPTTS	+5C97C	CLEAR	+5C981
CLEARC	+5C983	STRINK	+5C99A	RET4	+5C990	STXTPT	+5C99E
NETST	+5C998	LIST	+5C99C	STRTRNG	+5C9C8	ENDING	+5C9D8
MAINLIST	+5C9E3	KOLST	+5C9F1	LSTD?	+5CA0C	LSTLIN	+5CA0E

LISTLOOP=SCA15	SENDCHR =SCA19	SNCHR =SCA1C	NOR =SCA29
LISTRED =SCA38	GETCHR =SCA43	OC =SCA4B	TOKEN? =SCA4B
CONTOX =SCA5D	SKPTR =SCA6D	TAKLP =SCA6D	PTOK =SCA6A
TOKLUP =SCA6F	TOKNONE =SCA79	FOR =SCA8D	FOR? =SCA91
STEP =SCA89	ONESTEP =SCA8D	NEWSTT =SCA8C	DIRECT =SCAFF
TRACE? =SCB1F	EXECUTE =SCB37	GOEND =SCB4D	GOEND =SCB42
GOOAB2 =SCB44	GOOAB3 =SCB52	NOTOK =SCB5F	COLOR? =SCB62
JBY =SCB64	RESTORE =SCB69	SETDA =SCB73	RETS =SCB77
ISCHTC =SCB78	OK =SCB8C	ERFLG? =SCB8B	CTRC? =SCB91
STOP =SCB93	END =SCB95	END2 =SCB96	END3 =SCB4D
END4 =SCB4F	BOSTART =SCB8B	CONT =SCB8B	CON =SCB86
RETA =SCB04	SAVE =SCB05	LOAD =SCBEE	JLN =SCC12
WARTD =SCC15	PROG10 =SCC2A	RUN =SCC27	RUNLINE =SCC4D
GOOUB =SCC46	GOULNE =SCC5A	GOFO =SCC67	GAI =SCC7A
GO2 =SCC7E	RETT =SCC8F	PAP =SCC9D	UNDER? =SCCA1
OSTMER =SCC96	RETURN =SCC9F	DATA =SCCBA	ADON =SCCB0
RETB =SCCC7	DATAM =SCCC8	RENN =SCCC9	RM =SCCC3
RM2 =SCCDB	PULL3 =SCCDA	IF =SCCEE	TRIE? =SCCFD
REN =SCD01	IFTRUE =SCD06	JCRCHD =SCD0E	ANGOTO =SCD11
GOFO? =SCD019	INMT =SCD1D	WAKIM =SCD25	RETP =SCD3D
LINGET =SCD31	ASCHEX =SCD37	KACD0 =SCD65	LET =SCD68
LET2 =SCD88	LETREAL =SCD9C	LETSTR =SCD9F	PUSSTR =SCDAD
DESC? =SCD85	COSTR =SCD8F	MEMDESC =SCDCA	COPY =SCD8C
PRSTRNG =SCD94	PRINT =SCDFA	PRINT2 =SCDFC	CRD0 =SCD2B
NEGATE =SCD25	RETD =SCD27	TAB =SCD2B	MOCLN =SCD33
TAWHERE =SCD3B	SPC? =SCD46	TABIT =SCD5D	N3SPC =SCD51
NEXTCHR =SCD54	OUTPC =SCD5A	STROUT =SCD5F	STRFRT =SCD62
NICHAR =SCD69	OUTSP =SCD7C	OUTQUES =SCD7F	OUTD0 =SCD81
SEND =SCD89	INPUTERR =SCD96	READERR =SCDAD	ERLTM =SCD94
INPERM =SCD96	RESPERR =SCD9C	DIREENT =SCD95	GET =SCD9C
INPUT =SCD07	QOUT =SCD0F	OTR? =SCD0C	KKIM =SCD91
READ =SCD07	ZF =SCD0E	MAIDMP =SCD1D	KKIM? =SCD16
SNDR? =SCD47	STRP =SCD4F	INSTANT =SCD53	PUTCHR =SCD67
PENCHR =SCD73	FECHM =SCD74	SKP =SCD7F	MUMIN =SCD6B
DATIN =SCD91	MAX =SCD9A	SMPT =SCD96	INFFIN =SCDCL
FINDATA =SCD9B	N3S =SCD91	IMPONE =SCD9E	NTD =SCDFF
RETL1 =SCD96	EXTG =SCD07	REDIG =SCD17	NEXT =SCD21
WAKMT =SCD27	SKPV =SCD2B	GERR =SCD35	GOFFOR =SCD37
GOHWST =SCD7A	ENDFOR =SCD7D	FRRMUN =SCD8F	CHMUM =SCD92
CHSTR =SCD94	CHVAL =SCD95	RETI2 =SCD9B	CV2 =SCD9C
N3SHON =SCD9E	JERROR =SCDAD	FIREVL =SCDAD	ND1 =SCD9F
FEVL00P =SCD9E	FIREVL2 =SCD9D	CRDP =SCD9C	CHKTYP =SCD9C
AKTH =SCD9D	PREFTEST =SCD95	K3DP =SCD9E	SAVOP =SCD9F
COMPARE =SCD9C	MO2 =SCD116	PREFNC =SCD11E	PSMAD =SCD125
SAIXERR =SCD125	PSHF =SCD13B	PSHFACX =SCD13D	PUSHFAC =SCD14B
WTHMATH =SCD15D	GOEX =SCD16D	DONTH =SCD162	DONTH =SCD16B
DONMATH =SCD16D	EXIT =SCD185	GETVAL =SCD18B	SKTP =SCD18C
MURBER =SCD191	WAR? =SCD194	STRXT =SCD1A9	STL =SCD1B2
WOT? =SCD18B	EWOP =SCD1CD	WOT2 =SCD1C7	FW? =SCD1CC
SEN? =SCD1D3	PARCHK =SCD1DA	CHCKLS =SCD1ED	CHDFW =SCD1ED
CHCOM =SCD1E6	SYNCH =SCD1EB	SYNERR =SCD1F1	AKN =SCD1F6
COUL =SCD1F8	VARL =SCD1FD	VRL =SCD20D	VR2 =SCD21E
SCREEN =SCD221	UNARY =SCD234	NOTInstr =SCD262	GNDOUT =SCD267
ON =SCD277	AND =SCD27D	FALSE =SCD285	TRUE =SCD28B
POSWP =SCD2B0	STRCMP =SCD2B5	SFS =SCD2CD	CNLOOP =SCD2D2
MURCHP =SCD2B8	DOCMP =SCD2D0	CNPDME =SCD2E9	JF =SCD2F2
MODIN =SCD2F4	DIN =SCD2F9	PTARGET =SCD303	PTARGET2 =SCD30B

PTGET3	=S030A	BADNAM	=S0314	MANSE	=S0317	MOFMAN	=S0321
GTLY	=S032B	BTPASS	=S032C	STRNG?	=S0336	INTFAB?	=S0340
MM	=S0350	SCOOH	=S0357	JAR?	=S0360	BSS	=S0363
VSEARCH	=S0369	NSVM	=S0373	MV1	=S0375	MV2	=S037F
NPTFR	=S0380	ISLETC	=S0387	RTN1	=S0380	NOTFND	=S03A1
TWBERK	=S03B4	NEWAR	=S03B6	NW1	=S03CC	SETVPNT	=S03FB
SWP	=S0402	GETAR?	=S0407	GETAR?2	=S0409	GO	=S0413
HEGRUM	=S041B	MAXINT	=S041C	AKINT	=S0422	ALINT	=S0426
MS	=S0433	MS2	=S0435	ARRAY	=S043B	NOTDM	=S0446
FIDARY	=S04B3	ARFLD?	=S04B7	AR-NAME?	=S0493	ADARY	=S04A2
SUBERR	=S04B0	DAERR	=S04B3	JER	=S04B5	ARYFOUND	=S04B8
CHRO2M	=S04CA	NOTFNDM	=S04C2	MAKARY	=S04C8	NAME	=S04EF
RAR	=S04FB	SANDM	=S0501	OFLTOH	=S0511	ZARY	=S0534
ZLUP	=S0543	NOPE	=S054B	FINDEEN	=S05A5	ORMLJP	=S05B0
BSE	=S05B3	GME	=S05B6	OSMOK	=S05B9	OSMOK2	=S05BA
WDM	=S059B	WINTA	=S05AC	BARY	=S05B2	RTM2	=S05CA
MULT	=S05C7	HU1	=S05D0	MU2	=S05DA	MU3	=S05F3
FRE	=S05FB	FRE2	=S05FF	BIVANT	=S060C	POS	=S0619
SOMFLT	=S061B	ERRDOR	=S0620	UNOFNC	=S062B	DEF	=S063B
FNC?	=S065B	FUNCT	=S066E	SAWLD	=S0692	GETHLD	=S06C3
FNDATA	=S06C9	STR	=S06CF	STRINK	=S06EF	STRSPA	=S06F7
STRLIT	=S0701	STRLT2	=S0707	FEND	=S0711	QUO?	=S071E
ZEND	=S0722	HZ	=S0723	FEL	=S072F	FE2	=S0739
PUTHW	=S0744	JERR	=S074C	PUTENP	=S074F	GETSPA	=S076C
GETSPC	=S076E	CY	=S0779	OUTSPA	=S07B3	FULL	=S07BE
GRRAG	=S079E	FNDWAR	=S07A2	TWAR	=S07BA	SWMS	=S07C3
SWR	=S07CF	SWRBD	=S07B7	ARYWAR	=S07DC	ARYK2	=S07E4
ARYW3	=S07EB	ARYVGO	=S07F3	BPL1	=S0824	ARYSTR	=S0826
SL60	=S082E	DWMS	=S0833	DWMS	=S083D	DW1	=S0852
OW2	=S085C	OWMIS	=S086C	VDOME	=S0877	GRRPAS	=S087C
CAT	=S08B1	MTL	=S08B1	MOVING	=S08EE	MOVSTR	=S08FC
MOVESTR	=S0900	MWS2	=S0904	MWS3	=S0900	BPL2	=S0916
FRESTR	=S0917	FREFAC	=S091A	FRETWP	=S091E	BPL3	=S0949
MB	=S094A	FRETMS	=S094F	RTM3	=S095F	CHRSTR	=S0960
LEFTSTR	=S0974	DMS1	=S097A	DMS2	=S0990	DMS3	=S0991
DMS4	=S09B2	BPL4	=S0999	RIGHTSTR	=S09A0	WIGSTR	=S09A8
DMS5	=S09BC	INSTMS	=S09D3	LEN	=S09F0	GETSTR	=S09FF
ASC	=S09FF	DOT4	=S0A0C	GBRITC	=S0A0F	GETBYT	=S0A12
COMINT	=S0A15	VAL	=S0A21	VL2	=S0A29	VL3	=S0A41
POINT	=S0A57	GTUM	=S0A60	COMYTE	=S0A66	GETADR	=S0A6C
PEEX	=S0A7E	POXE	=S0A95	MLTB	=S0A9E	WT2	=S0AAB
WT3	=S0AB1	ETH4	=S0AB9	FAOBH	=S0ABA	FSOB	=S0AC1
FSOBT	=S0AC4	AD0	=S0AD3	FA0?	=S0AD8	FA0BT	=S0AD8
AD1	=S0AD8	AD2	=S0AEB	AD3	=S0B14	AD4	=S0B0B
AD5	=S0B14	SUBNAM	=S0B20	SGHIF	=S0B43	STGAF	=S0B4B
FLOOP	=S0B4C	ZEROFAC	=S0B6B	ALOFAC	=S0B6A	ALOFACS	=S0B6C
ADMAN	=S0B6F	FR1	=S0B8E	FR2	=S0B96	FR3	=S0B87
FBOMG	=S0B89	RTMS	=S0B87	NEGFAC	=S0B8B	MEG2	=S0B8E
PLUSFP	=S0BFD	RTM6	=S0BEE	OVERFLOW	=S0BEF	SHFTRES	=S0BF4
MSFT	=S0BF4	SHOFT	=S0C0A	SH1	=S0C17	SH2	=S0C19
SHFT	=S0C21	SH3	=S0C2B	MUM	=S0C2B	LOGSER	=S0C32
SHFTWIS	=S0C47	SMDMS	=S0C4C	Freebie	=S0C51	LOGMS	=S0C56
LOG	=S0C5B	OTB	=S0C62	LG2	=S0C65	FMULT	=S0C99
FMULT	=S0C9C	FMU	=S0CA1	FMU	=S0CCA	FR2	=S0CDE
FMG	=S0CDE	FMH	=S0CEE	RTM?	=S0CFC	COMPK	=S0CFF
NOEX?	=S0CEB	NOEX2	=S0CEB	NOEX3	=S0CCE	NOEX4	=S0CEB
OUTOFNG	=S0CEB	ZERO	=S0CEB	JOU	=S0CCE	MULT0	=S0CEB

RTM8	=S0049	MUM10	=S006A	DIV10	=S006F	DIV	=S0078
ENDWIMP	=S007A	FDIV	=S0080	FDIVT	=S0083	FD1	=S009A
2 F08	=S00A2	F02	=S0080	F03	=S0080	F04	=S00C0
F05	=S00CE	F06	=S00E8	F07	=S00EF	DIV2	=S00F8
RESIFAC	=S00E0	NOVFN	=S0E13	NOV2F	=S0E38	NOV1F	=S0E38
2 NOVNL	=S0E20	SETFOR	=S0E41	NOVNF	=S0E45	NOVFA	=S0E60
NFA	=S0E6F	NFA2	=S0E73	NOVNF	=S0E70	NMF	=S0E80
NMF2	=S0E82	RTM9	=S0E88	RM08	=S0E8C	RM080	=S0E9A
SDM	=S0E9C	SDM1	=S0EAD	SDM2	=S0EAD	RTM10	=S0EAD
SGH	=S0EAA	FLOAT	=S0EAD	FL01	=S0EB5	FL02	=S0E8A
AMS	=S0EC9	FDMF	=S0EDC	FDMF2	=S0EDC	FC1	=S0F03
FC2	=S0F09	GINT	=S0FDC	G11	=S0F20	RTM11	=S0F28
G12	=S0F2C	INT	=S0F30	ZFAC	=S0F5A	RTM12	=S0F63
FTN	=S0F34	FM2	=S0F60	FM3	=S0F77	EVNL	=S0F78
NOOISIT	=S0F7E	CMOP	=S0FB0	SETSEM	=S0F9F	DP01G	=S0FA1
SGEX	=S0FA4	SEMOM	=S0FA6	SETDP	=S0FB2	ADJEXP	=S0FB8
NEC	=S0F8A	DPLEFT	=S0FC3	DPATHT	=S0FCC	EVDOE	=S0FD3
EVO	=S0FB8	THSRTD	=S0FD8	NOP	=S0FE2	ADACC	=S0FEF
GETEXP	=S0ED2	MW06	=S0E11	STEX	=S0E1F	CENWEDC	=S0F04
RTLWEDM	=S0E29	BLWAD	=S0ED2	IMPT	=S0ED3	LWPT	=S0E0E
2 PRINTFAC	=S0E48	PRSTR	=S0E48	FOUT	=S0E4E	FACSTRMG	=S0E50
SFSG	=S0E58	NOTZE	=S0E69	MI	=S0E71	STE	=S0E7A
CMFNI	=S0E7C	CMFNI	=S0E87	JM10	=S0E82	JC10	=S0E89
ROUW	=S0EAD	INTPART	=S0EAD	OPLOC	=S0E8B	OPFL	=S0E89
PUDOP	=S0EC4	SUY	=S0E05	MAVSTR	=S0E87	MSUP	=S0E8E
PARITY?	=S0EFF	COUNTED	=S0E10	MAVOTSET	=S0E10B	SAVY	=S0E123
MWACK	=S0E133	NEEDER	=S0E140	PUTEA	=S0E150	MAVTK	=S0E15C
WRUP	=S0E171	HARKEND	=S0E174	PHYSTK	=S0E179	HALF	=S0E17E
DECTBL	=S0E183	2 TEND	=S0E1A7	SAR	=S0E1A7	FWART	=S0E1B1
PW1	=S0E18A	PW2	=S0E1D4	REGOP	=S0E1EA	RTM13	=S0E1FA
LOGA	=S0E1F5	EXPSER	=S0E1FA	EXP	=S0E223	X1	=S0E233
208	=S0E23E	X2	=S0E241	X3	=S0E251	COOSER	=S0E276
SERIES	=S0E28C	SERMA2M	=S0E290	S5	=S0E29F	SERLADP	=S0E2A3
WATERM	=S0E280	RTM14	=S0E28F	RMADJ1	=S0E2C0	RMADJ2	=S0E2CA
RMD	=S0E2C8	RD1	=S0E2E6	RD2	=S0E301	COS	=S0E304
SIM	=S0E308	S11	=S0E330	S12	=S0E340	S13	=S0E340
TAM	=S0E354	TAK2	=S0E37C	P1div2	=S0E380	P1vack2	=S0E382
QUARTER	=S0E38A	SIMSER	=S0E38F	ATH	=S0E38B	ATH1	=S0E3C0
ATM2	=S0E3CE	ATM3	=S0E3E1	RTM15	=S0E3E7	ATMSER	=S0E3E3
ZFSTUFF	=S0E425	N10	=S0E42B	RTM16	=S0E43C	COLOST	=S0E442
MVZP	=S0E46C	TABK	=S0E44F	SET48K	=S0E4C1	HISFFY	=S0E4C7
STDMEN	=S0E4D1	IM1080	=S0E4E7	IM18A5	=S0E4E9	MWASLOC	=S0E4FA
IM18A53	=S0E51A	CALL	=S0E535	PLSTFMS	=S0E540	GOERL	=S0E55A
LINC308	=S0E55D	AT?	=S0E5AC	PLOT	=S0E579	MLTN	=S0E58A
VLIN	=S0E595	COLON	=S0E5A3	EVNLVAR	=S0E5AA	STCOR	=S0E59F
VTAB	=S0E580	SPEED	=S0E5C9	TRACE	=S0E5DA	NOTRACE	=S0E5DA
NORMAL	=S0E5DA	INVERSE	=S0E5D0	2 WRM	=S0E5DF	2 S1	=S0E5E1
HDMEN	=S0E5E7	JRM	=S0E5F7	SETHI	=S0E5FA	LOMEN	=S0E607
QWERR	=S0E62C	MMADLER	=S0E64A	RESUME	=S0E67F	JSTM	=S0E68F
DEL	=S0E6P2	ML	=S0E68B	MOVDM	=S0E6CA	MOVDT	=S0E6CB
M02	=S0E602	M03	=S0E60B	M04	=S0E6E9	GRI	=S0E6F1
OR	=S0E6F9	TEXT	=S0E703	STORE	=S0E70F	SL1	=S0E71F
RECALL	=S0E70C	M04	=S0E747	HGR2	=S0E74B	2 HGR1	=S0E74F
HGR	=S0E75B	HTTAT	=S0E764	2 SETAPG	=S0E769	2 HGR	=S0E771
2 HGR0	=S0E775	HGR01	=S0E77D	HP05H	=S0E79D	HP05H1	=S0E7C1
HP05M2	=S0E7C2	HPLOD	=S0E7D7	LFTRT	=S0E7E5	2 LEFT	=S0E7E7
LRI	=S0E7EE	LEFT1	=S0E7F1	LEFT2	=S0E7F6	MEMM1	=S0E7F8

2	CSHIFT	=SE7FC	CSHFT2	=SE7FE	RTS1	=SE809	RIGHT	=SE80A
	LRUD01	=SE80C	LRUD02	=SE810	LRUD1	=SE833	LRUD2	=SE834
	LRUD3	=SE844	LRUD4	=SE848	ERG	=SE840	UPDOWN	=SE853
2	UP0	=SE855	UP1	=SE868	UP5	=SE874	UP2	=SE878
	UP3	=SE87D	UP4	=SE87F	UPDOWN1	=SE881	DOWN	=SE885
	EQ4	=SE888	DQW03	=SE8A4	DQW04	=SE8A8	DQW05	=SE8AD
2	HLINRL	=SE8B1	HLIN	=SE8B8	HLIN2	=SE8D1	HLIN3	=SE8E9
	MOVEX	=SE8FD	MOVEX2	=SE9D2	MOONIT	=SE9DC	HLIN4	=SE91F
	MSK7BL	=SE9F3	EQ1C	=SE93A	CGSTBL	=SE938	HFIND	=SE94C
	HFIND1	=SE971	HFIND2	=SE97F	OTS2	=SE9B1	ORAW0	=SE982
	DRAW1	=SE986	DRAW2	=SE9A7	ORAW3	=SE9B1	ORAW1	=SE98C
	ORAW5	=SE9C9	RG4	=SE9D9	2	XORAW0	XORAW1	=SE9E2
	XORAW2	=SEAD0	XORAW3	=SEAD0	XORAW4	=SEAE1A	XORAW5	=SEAE5
	HCS	=SEAE5	HTWS	=SEAE3A	HFMS1	=SEAE4E	DEGER	=SEAE7
	MCOLOR	=SEAGA	EVVAR	=SEAF1	RTS3	=SEAFD	HPLOT	=SEAE7E
	HP2	=SEAB8	HP3	=SEABF	ROT	=SEAM1	SCALE	=SEAM7
	DRAWNT	=SEAD0	OP1	=SEAC1	DP2	=SEAC7	DP3	=SEAE6
	ORAW	=SEAE9	XORAW	=SEAEF	SHLDAD	=SEAF5	SL1	=SEB16
	SL2	=SEB2D	SL3	=SEB23	TAPEFMT	=SEB3C	GETARVP1	=SEB59
	HTAB	=SEB67	HTAB1	=SEB4C	HTAB2	=SEB7A	NOTOR	=SEB7D
	AORW?	=SEB87	NOTRA	=SEB8D	MLD0	=SEB92	CH2FLT	=SEB9E
	MSAVE	=SEBC4	CH2FST	=SEBDD	CHKT	=SEBEE	JSTMER0	=SEBF2
	AORW0	=SEBF5	MP	=SEBF8	MA	=SEBEO	CONTS	=SEBDA
	LSVD	=SEB15	DSBFF?	=SEB2C	CHERF	=SEB3A	TSRDT?	=SEB48
	NTRET	=SEB38	ISCTH?	=SEB40	ISCTH	=SEB7D	PARREG	=SEB79
	TESFLO	=SEBDA	SETA42	=SEB92	SAVET	=SEB9A	IFORW	=SEBDAE
	NOZEROD	=SEBDC	NOZSAV	=SEBCC7	NOZERD1	=SEBCC	FMSAVE	=SEBDD
	PRP	=SEB19	WPRH	=SEB2A	CSST1	=SEBDD	RETN6	=SEBDD5
	CSST2	=SEB51	CSST3	=SEB5A	CSST4	=SEB73	CSST5	=SEB7C
	CSST6	=SEB8C	CSST7	=SEB95	RETN2	=SEB9E	TFM	=SEBDF
	NRH	=SEDA1	LOADT	=SEDAE	MOULNH	=SEBDD	INCNUM	=SEBDC3
	PGMMH	=SEDEA	CPLTRIT	=SEDDC	LECL	=SEDDC	OKCSUM	=SEBDB7
	LECHN	=SEED0	INFMMH	=SEED1	CPLTNRH	=SEEDA	IMPSP	=SEED45
	CONFNAME	=SEED3	INFMM1T	=SEED7	ATLCOP	=SEED7E	RETN3	=SEEDB7
	TOSUM1	=SEED8	ACOMPAL	=SEED9	WFFP	=SEEA8	STMP	=SEEDB
	ACOMPAS	=SEEB9	ACOMP#2	=SEEC7	PHAD0R	=SEED3	TSB	=SEEDDE
2	TOSUM	=SEED0	ZND01	=SEEC7	ZND0	=SEED9	KWORD1	=SEEDDE
	KWORD	=SEEDF	NOVSP	=SEEDF	KICHMMH	=SEEDF	CMLSP	=SEED18
	ZDRFNAME	=SEF1C	FIMMMH	=SEF28	RUSCOR	=SEF28	EXT101	=SEF33
	LDPP20	=SEF44	FRESP	=SEF49	FDESP	=SEF57	TENCAT	=SEF59
	ZTSUTL	=SEF77	CARDF	=SEF7E	TEPRAL	=SEF84	NRVIO	=SEF91
	ISCTP	=SEF9A	TSFOWL	=SEFA2	MOV001	=SEFA0	SVD0	=SEF82
	RECAP	=SEFC0	PL0T1	=SEFC8	PL0T0	=SEFCA	DESIMV	=SEFED
	KIMASER	=SEFEF	LINCOR0	=SEFF1	GRPL0T	=SEF001	TH0	=SEF04
	INA	=SEF08	SAVS	=SEF0D	PROCTCL	=SEF02	ATROUC	=SEF02A
	WQMC	=SEF032	TESTCL	=SEF03D	NOX	=SEF03D	SCANL	=SEF043
	HAMLTN	=SEF054	NOCOL	=SEF05E	NOKEY	=SEF064	SAVIX	=SEF07D
2	CLURKEY	=SEF074	RT1	=SEF07E	RT2	=SEF07F	IDFCTL	=SEF081
	SETCTL	=SEF09D	SCMD0	=SEF0B1	PCMD0	=SEF0B4	PCMD1	=SEF0B8
	CH2TXT	=SEFDD0	TEXT0	=SEFDDC	PUSCHR	=SEFDDC	DPCH	=SEF0E9
	FIM1	=SEF10A	ZPGA	=SEF10C	2	RFPGA	RFPGA	=SEF11D
	JPCR	=SEF127	TSAA6	=SEF12A	JCAIT6	=SEF133	SCROLL	=SEF134
	SCRL1	=SEF140	SCRL2	=SEF14A	SCRL3	=SEF171	SCRL4	=SEF173
	CLURL	=SEF198	OUT2	=SEF1A6	IOCTP	=SEF1A9	NOTCP	=SEF1C2
	SETER	=SEF1C7	CMTCR	=SEF1CA	DECTY	=SEF1D0	ESFERA	=SEF1E3
	JCAR19	=SEF1F5	JPR018	=SEF1F9	GERCHR	=SEF200	GERDRF	=SEF40D
	TABCSOR	=SEF47D	DISK10	=SEF59D	TITUL0	=SEF591	TABAL	=SEF5D0

ZKX	=SF65D	PROXIN	=SF661	CARACT	=SF67F	BOIRAF	=SF691
STODNG	=SF69C	ZERNFLD	=SF6A7	TRANSBUF	=SF6B0	FINTFGW6	=SF6B0
PLUBAS	=SF6B8E	OUTRCH	=SF6C1	LSCHIR	=SF6C0	EDTBAG	=SF6D4
THOB8	=SF6D8A	CONTB	=SF6F3	FIMPA	=SF6FC	FENBAS	=SF706
ENDTB8	=SF707	INOBAS	=SF709	BUSTCEER	=SF73G	DECACTW	=SF741
FLKAD16	=SF744	ETOKEN	=SF753	CONVTON	=SF75C	IGCBAS	=SF769
JPRSC	=SF77A	IMPRAS	=SF770	PALETR	=SF781	ULTIV	=SF79F
FFCRET	=SF798	JOLTRUF	=SF79F	JPRSC2	=SF78A	VTBFE	=SF7B0
NETCHBA	=SF7C8	FIMDEC	=SF7D2	OC18	=SF7DC	JDECBUF	=SF7E7
CELEFAX	=SF7EA	RTCR	=SF7FA	RTCS	=SF7F7	GHMOU10	=SF7F8
PLQ72	=SF800	PLQTH	=SF803	UPBLOC	=SF812	NLIME	=SF819
NLINE1	=SF83C	NLINE2	=SF826	NLINE	=SF828	RETN4	=SF831
CLURSC	=SF832	CLURTOP	=SF836	CLRSC2	=SF838	CLPSC3	=SF83C
GRABCALC	=SF847	DECALC	=SF856	? NEXTCOL	=SF85F	SETECL	=SF864
SCRN	=SF869	PWCOC	=SF874	PCCC	=SF8D7	INSDJ1	=SF885
INSD52	=SF88F	INSD53	=SF891	JEVEN	=SF89E	ERR	=SF8AB
GETFMT	=SF8AC	NNNDX1	=SF8C1	NNNDX2	=SF8C3	NNNDX3	=SF8CC
INSTD3P	=SF900	PRNTOP	=SF8D4	PRNTBL	=SF8D8	PRNM1	=SF8F5
PRNK2	=SF8F9	PRADR1	=SF910	PRADR2	=SF914	PRADR3	=SF924
PRADR4	=SF92A	PRADR5	=SF930	RELADR	=SF938	PRMTYX	=SF940
PRMTAX	=SF941	PRMTX	=SF944	PRELWK	=SF948	PREL2	=SF94A
PRBL3	=SF94C	PCADJ	=SF953	PCADJ2	=SF954	PCADJ3	=SF956
PCADJ4	=SF95C	REIMS	=SF961	FN1	=SF962	FN2	=SF965
CHNR1	=SF98A	CHNR2	=SF98A	MNERL	=SF9C0	MNER	=SF9D0
IR6	=SF9A0	BREAK	=SF9AC	XBRK	=SF9A9	RESET	=SF9A2
SCRN2	=SF9A5	RTHSKZ	=SF9A8	RSETH1	=SF9A6	FIXSEV	=SF9A4
EXIT0	=SF9AF	? NOFIX	=SF9A2	PPREG	=SF9AC	PARUP	=SF9AB
SETPLP	=SF9AD	? SLOPP	=SF9AC	WRTHYT	=SF9AB	RESDSP	=SF9AC
RODSP1	=SF9AF	ROSP1	=SF9A9	MEMS	=SF9AF3	INC	=SF9B2
SEPTAB	=SF9B5	WTCTL	=SF9B14	KITPOS	=SF9B20	NOTPOS1	=SF9B5
JCODCR	=SF9B2E	POSES9	=SF9B31	SMIT1	=SF9B38	SETMNO	=SF9B4C
TABV	=SF9B5C	OPTX2000	=SF9B5F	STITLE	=SF9B68	VIDWAIT	=SF9B70
NPSCH1	=SF9B8A	LOKET	=SF9B93	MOWAIT	=SF9B95	VISTR1	=SF9B96
SAVEL	=SF9B90	SMZ	=SF9B9F	TABCOR	=SF9BA8	POSCUR	=SF9BB2
IMVOR	=SF9BC4	BELL0	=SF9BD9	? BELL1	=SF9BD0	BELL2	=SF9BE4
RTSCB	=SF9BF	WFOG1	=SF9BD0	CVX8	=SF9BF3	AUX	=SF9BF5
IOFCOD	=SF9FD0	? VIDOUT	=SF9C05	DESESB	=SF9C18	DESESB1	=SF9C20
UP	=SF9C23	RTSA	=SF9C28	RTSA	=SF9C2C	PRRESB	=SF9C20
IMS	=SF9C30	RSTR1	=SF9C3A	? CLNEOP	=SF9C42	CLEOP1	=SF9C46
HONE	=SF9C38	CR	=SF9C42	LF	=SF9C44	EPOVID	=SF9C73
CLERL	=SF9C99	CLEOL2	=SF9C98	CLEREP	=SF9C90	WAIT	=SF9C48
WAIT2	=SF9CA9	WHIT3	=SF9CA4	NOTA4	=SF9CB4	NEXTAL	=SF9C8A
RTS48	=SF9C18	HEADR	=SF9CC9	MRE11	=SF9CD6	ZERDLY	=SF9C88
QHEOLY	=SF9CE2	WTAPE	=SF9CE5	RDYITE	=SF9CEC	RDYITE2	=SF9CE8
RD2B17	=SF9CFA	ROB17	=SF9CFD	TNO	=SF9CFC	TNI	=SF9D0F
CLRES9	=SF9D12	DECBUF	=SF9D18	SRZCUR	=SF9D25	? INTC	=SF9D35
PSCR	=SF9D38	POSCUR1	=SF9D30	BRGOUT	=SF9D54	SRGUR	=SF9D58
SRGUT	=SF9D58	JPRRESB	=SF9D64	GETLWZ	=SF9D67	GETLW	=SF9D68
? GETLW1	=SF9D4F	CLURIN	=SF9D73	JWKS	=SF9D7C	VFLW	=SF9D82
? ADDIMP	=SF9D86	? CROUT1	=SF9D88	CROUT	=SF9D8E	PRM1	=SF9D92
PRYX2	=SF9D96	XANO	=SF9D93	NOBCHX	=SF9DAD	XAN	=SF9D83
DATAGUT	=SF9D86	? RTSAC	=SF9DC5	XANPH	=SF9DC6	? TSTOP	=SF9DCA
NOO	=SF9D01	PRRYTE	=SF9D6A	? PRMEX	=SF9DC3	PRMEX2	=SF9DC5
COUT	=SF9D6D	COUT1	=SF9D6D	MENTAB	=SF9DF3	SAVEADR	=SF9D8E
? LEADADR	=SF9D7D	BL1	=SF9D00	? BLANK	=SF9ED4	? STGR	=SF9ED8
RTS5	=SF9E17	? SETAND	=SF9E18	SETAND2	=SF9E1D	? LT	=SF9ED0
L72	=SF9E22	NOVE	=SF9E2C	V7Y	=SF9E3A	VF7OK	=SF9E30

? NLIST	=SFE5E	LIST2	=SFE6D	ALPC	=SFE75	ALPCLP	=SFE7D
ALPCATS	=SFE7F	? SETENV	=SFE8D	SETMWRM	=SFE8A	SETIFLG	=SFE86
SETKBC	=SFE89	? IMPRT	=SFE8B	? NIMPRT	=SFE8D	SETVID	=SFE93
? OUTPMT	=SFE95	? OUTPRT	=SFE97	IOPT	=SFE9B	IOPT1	=SFEA7
IOPT2	=SFEA9	? XBASIC	=SFE8D	? BASCONT	=SFE83	GO	=SFEBA
? REG2	=SFE8F	? TNET	=SFE82	? TINET	=SFE84	RTOL	=SFE85
? USER	=SFE84	WRITE	=SFE8D	WR1	=SFE84	WRBITE	=SFE8D
WRBYT2	=SFE8F	? CMMN	=SFE86	READN	=SFE8D	READ2	=SFE82
WRD2	=SFE8A	RD3	=SFE86	PRERR	=SFE8D	BELL	=SFE8A
IORESTORE	=SFE8F	? IOSAVE	=SFE8A	SAVD	=SFE8D	MM	=SFE8D
MMZ	=SFE86	ZWDD	=SFE8C	WRTITH	=SFE8F	CHMSCH	=SFE76
CLREOL1	=SFE86	DIG	=SFE8D	NXTBIT	=SFE93	NXTBAS	=SFE9B
NXTBAS2	=SFE8F	GETNUM	=SFE8A	NATCHR	=SFE8D	NACCD	=SFE8D
COBNTS	=SFE89	SNTEL	=SFE8A	SUBTBL	=SFE8E		

BIBLIOGRAFIA

Brown, P. J. - Writing Interactive Compilers and Interpreters. 3a ed., New York, John Wiley & Sons, [1981].

Leventhal, Lance A. - 6502 Assembly Language Programming. 1a ed., Berkeley, Osborne/McGraw-Hill, [1979].

McIntire, Thomas C. - Software Interpreters for Microcomputers. 1a ed., New York, John Wiley & Sons, [1978].

Microdigital Eletrônica Ltda. - Manual Técnico do TK 2000 Color.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------